PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Kyung-Hee LEE et al. Docket: 678-1395 (P10801)

Serial No.: 10/800,181 Dated: April 19, 2004

Filed: March 12, 2004

For: **APPARATUS AND METHOD FOR PERFORMING MONTGOMERY TYPE MODULAR MULTIPLICATION**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF PRIORITY DOCUMENT

Sir:

Enclosed is a certified copy of Korean Appln. No. 2003-16100 filed on March 14, 2003 from which priority is claimed under 35 U.S.C. §119.

Respectfully submitted,

Paul J. Farrell
Registration No. 33,494
Attorney for Applicants

DILWORTH & BARRESE, LLP
333 Earle Ovington Boulevard
Uniondale, New York 11553
(516) 228-8484

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8 (a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on April 19, 2004.

Dated: April 19, 2004

Paul J. Farrell

대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2003-0016100
Application Number

출 원 년 월 일 : 2003년 03월 14일
Date of Application MAR 14, 2003

출 원 인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2004 년 03 월 12 일

특 허 청
COMMISSIONER

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0003
【제출일자】	2003.03.14
【국제특허분류】	G06F
【발명의 명칭】	몽고메리 유형의 모듈라 곱셈 장치 및 방법
【발명의 영문명칭】	APPARATUS AND METHOD FOR PERFORMING A MONTGOMERY TYPE MODULAR MULTIPLICATION
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	2003-001449-1
【발명자】	
【성명의 국문표기】	이경희
【성명의 영문표기】	LEE,Kyung Hee
【주민등록번호】	710409-1648917
【우편번호】	449-904
【주소】	경기도 용인시 기흥읍 보라리 쌍용아파트 113-902
【국적】	KR
【발명자】	
【성명의 국문표기】	임범진
【성명의 영문표기】	IM,Bum Jin
【주민등록번호】	750926-1052120
【우편번호】	442-372
【주소】	경기도 수원시 팔달구 매탄2동 198-23번지 201호
【국적】	KR
【발명자】	
【성명의 국문표기】	허미숙
【성명의 영문표기】	HUH,Mi Suk

1020030016100 출력 일자: 2004/3/24

【주민등록번호】	671125-2011913
【우편번호】	135-080
【주소】	서울특별시 강남구 역삼동 834-18
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 이건주 (인)

【수수료】

【기본출원료】	20	면	29,000 원
【가산출원료】	50	면	50,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	23	항	845,000 원
【합계】	924,000 원		

【요약서】

【요약】

스마트 카드 및 이동 단말기와 같은 통신시스템의 환경에서 고속의 암호화/복호화 및 전자서명을 위한 모듈라 곱셈 장치가 개시되어 있다. 이러한 본 발명은 n비트의 승수(A)와 피승수(B)를 입력으로 하고 (M+2)클럭(여기서, $m=n/2$) 내에서 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하기 위하여 몽고메리 유형의 모듈라 곱셈 연산을 수행하는 장치를 제안한다. 이러한 제안에 따른 장치들은 승수의 비트들을 순차적으로 쉬프트시켜 쉬프트된 비트열을 생성하고, 상기 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하는 것을 특징으로 한다. 이에 따라 본 발명은 모듈라 곱셈 장치가 적은 게이트 수를 가지도록 하고, 저전력을 소모하도록 하고, 고속으로 동작하도록 한다.

【대표도】

도 1

【색인어】

모듈라 곱셈, 몽고메리 유형, 기록 로직, 고속 동작

【명세서】

【발명의 명칭】

몽고메리 유형의 모듈라 곱셈 장치 및 방법 {APPARATUS AND METHOD FOR PERFORMING A MONTGOMERY TYPE MODULAR MULTIPLICATION}

【도면의 간단한 설명】

도 1은 본 발명의 제1 실시예에 따른 모듈라 곱셈 장치의 구성을 보여주는 도면.

도 2는 도 1에 도시된 기록 로직의 보다 구체적인 구성을 보여주는 도면.

도 3은 도 1에 도시된 캐리저장형 가산기(CSA)1의 보다 구체적인 구성을 보여주는 도면.

도 4는 도 1에 도시된 몫 로직의 보다 구체적인 구성을 보여주는 도면.

도 5는 도 1에 도시된 CSA2의 보다 구체적인 구성을 보여주는 도면.

도 6은 도 1에 도시된 전가산기(FA)의 보다 구체적인 구성을 보여주는 도면.

도 7은 본 발명의 제2 실시예에 따른 모듈라 곱셈 장치의 구성을 보여주는 도면.

도 8은 도 7에 도시된 기록 로직의 보다 구체적인 구성을 보여주는 도면.

도 9는 도 7에 도시된 CSA1의 보다 구체적인 구성을 보여주는 도면.

도 10은 도 7에 도시된 몫 로직의 보다 구체적인 구성을 보여주는 도면.

도 11은 도 7에 도시된 CSA2의 보다 구체적인 구성을 보여주는 도면.

도 12는 도 7에 도시된 FA의 보다 구체적인 구성을 보여주는 도면.

도 13은 본 발명의 실시예에 따른 모듈라 곱셈 장치들의 적용 예를 보여주는 도면.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<14> 본 발명은 암호화 기술에 관한 것으로, 특히 암호화/복호화 및 디지털 서명 등에 사용하기 위한 몽고메리 유형의 모듈라 곱셈 장치 및 방법에 관한 것이다.

<15> 전자 상거래를 위한 스마트 카드(smart card) 및 전자 화폐, 셀룰라 전화기와 같은 이동 통신기기(mobile device), 소형 컴퓨터(small-sized computer) 등을 사용하는 통신시스템에서 정보(전자적 문서 혹은 데이터)를 암호화/복호화하거나 전송 정보에 디지털 서명을 행하여 안전하게 전송하는 것은 바람직한 일이다. 여기서 디지털 서명이라 함은 정보의 전자적인 교환에 있어서 종래 종이에 자필 서명한 기능을 전자적 문서에 그 기능을 제공할 수 있도록 하는 기술을 의미한다. 특히, 최근에 인터넷의 사용 인구가 급속하게 증가하고 또한 상기 인터넷을 통한 개인 정보 등의 전송이 빈번해짐에 따라 안전하지 못한 채널을 통한 보다 안전한 정보의 전송이 요구되고 있는 실정이다.

<16> 공개키 방식 등을 이용한 암호화/복호화 및 디지털 서명에는 RSA(Rivest-Shamir-Adleman), ElGamal, Schnorr 등 여러 제안된 알고리즘이 사용될 수 있다.

상기 알고리즘의 국제 표준으로는 RSA 알고리즘에 기반한 ISO(International Standard Organization)/IEC(International Electrotechnical Commission) 9796이 채택되었으며, 미국에서는 ElGamal의 변형으로 DSA(Digital Signature Standard)가 채택되었으며, 러시아에서는 러시아연방국가 표준(GOSSTANDART: 통상 GOST라고 하는)에 따른 방식이 채택되었으며, 한국에서는 KC-DSA가 채택되었다. 이와 달리, 실제 여러 통신시스템들에서는 많은 PKCS(Public Key Cryptography Standard) 표준이 구현되고 있는 실정이다. 전술한 알고리즘들을 위해서는 모듈라 멱승인 $m^e \bmod N$ 연산이 요구되는데, 이 모듈라 멱승은 모듈라 곱셈, $A \cdot B \bmod N$ 연산의 반복적인 수행을 의미한다.

<17> RSA 등 공개키 암호에 기반한 디지털 서명을 생성 및 검증하는데 필수적인 모듈라 곱셈 연산을 수행하기 위한 많은 알고리즘들이 제안되었다. 예를 들어, R. L. Rivest et al, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, Vol. 21, pp. 120-126, 1978, P. L. Montgomery, "Modular Multiplication without Trial Division," Math. Of Comp., Vol. 44, No. 170, pp. 519-521, 1985, S. R. Dusse and B. S. Kaliski Jr., "A cryptographic library for the Motorola DSP56000, " Proc. Eurocrypto'90, pp. 230-244, 199. Springer-Verlag, A. Bosselaers, R. Govaerts and J. Vandewalle, "Comparison of three modular reduction functions," Advances in Cryptology - CRYPTO'93, pp. 175-186, 1993과 같은 논문들이 있다. 상기 알고리즘들 중 몽고메리(Montgomery) 알고리즘은 단순 모듈라 곱셈에서는 번거로운 절차를 가지고 있으나, 다양한 암호 알고리즘에 필요한 모듈라 멱승을 위한 모듈라 곱셈에는 계산 효율면에 있어서 가장 효율적이라는 사실이 논문 D. R. Stinson, Cryptography, CRC Press, 1995에 밝혀졌다. 미합중국 특

허 US Patent No. 6,185,596은 상기 Montgomery 알고리즘에 의해 구현한 장치의 일 예를 보여주고 있다.

<18> 전술한 바와 같이 공개키 암호화/복호화 및 전자 서명을 위한 많은 알고리즘 및 아키텍처가 제안되었다. 그러나 대부분의 제안된 알고리즘 및 아키텍처들에 따른 모듈라 곱셈 장치는 고속의 공개키 암호화/복호화를 목적으로 한 것이기 때문에 많은 수의 게이트(Gate) 들을 필요로 하고, 많은 전력을 소모한다는 단점이 있다.

【발명이 이루고자 하는 기술적 과제】

<19> 따라서 본 발명의 목적은 스마트 카드 및 이동 단말기와 같은 통신시스템의 환경에서 고속의 암호화/복호화 및 전자서명을 위한 모듈라 곱셈 장치가 적은 게이트 수를 가지도록 하는 데 있다.

<20> 본 발명의 다른 목적은 스마트 카드 및 이동 단말기와 같은 통신시스템의 환경에서 고속의 암호화/복호화 및 전자서명을 위한 모듈라 곱셈 장치가 저전력 소모를 가지도록 하는 데 있다.

<21> 본 발명의 또 다른 목적은 스마트 카드 및 이동 단말기와 같은 통신시스템의 환경에서 암호화/복호화 및 전자서명을 위한 모듈라 곱셈 장치가 고속으로 동작하도록 하는 데 있다.

<22> 이러한 목적들을 달성하기 위한 본 발명은 n비트의 승수(A)와 피승수(B)를

입력으로 하고 $(n+2)$클럭(여기서, $m=n/2$) 내에서 $A\cdot B\cdot R^{-1} \bmod N$ (여기서, R 4^{m+2})을 계산하기 위하여 몽고메리 유형의 모듈라 곱셈 연산을 수행하는 장치 및 방법을 제안하며, 이러한 제안에 따른 장치들은 승수의 비트들을 순차적으로 쉬프트시켜 쉬프트된 비트열을 생성하고, 상기 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하는 것을 특징으로 한다.

<23> 본 발명의 제1 실시예에 따르면, 모듈라 곱셈 연산 장치의 기록 로직은 상기 승수의 비트들이 순차적으로 쉬프트되어 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 상기 피승수를 다중화하여 $(n+4)$비트의 부호있는 이진 수들을 출력한다. 제1 캐리저장형 가산기(CSA)는 $(n+4)$개의 전가산기들을 구비하고, $(n+2)$비트의 제1 신호와, $(n+3)$비트의 제2 신호와, 상기 기록 로직으로부터의 $(n+4)$비트의 상기 이진 수들을 제3 신호로 입력하고, 상기 제1 신호의 상위 $(n+2)$번째 비트는 상기 전가산기들중 상위 3개의 전가산기들로 입력되고, 상기 제2 신호의 상위 $(n+3)$번째 비트는 상기 전가산기들중 상위 2개의 전가산기들로 입력되고, 상기 $(n+3)$개의 전가산기들에 의해 $(n+4)$비트의 캐리 값들과 합 값들을 출력한다. 몫 로직은 상기 제1 CSA로부터의 상기 $(n+4)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들과 하위 1개의 전가산기로부터 출력되는 캐리 값을 입력하고, 모듈라 감소의 배수를 결정하기 위한 3비트의 결정 값을 출력한다. 선택기는 상기 결정 값에 따라 미리 정해진 모듈로 수의 집합중 하나의 모듈로 수를 선택하여 출력한다. 제2 캐리저장형 가산기(CSA)는 $(n+4)$개의 전가산기들을 구비하고, 상기 선택기로부터의 선택된 $(n+4)$비트의 모듈로 수를 제1 신호로서 입력하고, 상기 제1 CSA로부터의 상기 $(n+4)$비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 $(n+3)$비트의 캐리 값들을 제2 신호로서 입력하고, 상기 $(n+4)$비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 $(n+3)$비트의

합 값들을 제3 신호로서 입력하고, 상기 제1 신호의 $(n+4)$비트들은 상기 전가산기들의 최하위 전가산기들로부터 순차 입력되고, 상기 제2 신호의 $(n+3)$비트들은 상기 전가산기들중 하위 2번째 전가산기들로부터 순차 입력되고, 상기 제3 신호의 $(n+3)$비트들은 상기 전가산기들의 하위 2번째 전가산기들로부터 순차 입력되고, 상기 $(n+4)$개의 전가산기들에 의해 $(n+4)$비트의 캐리 값들과 합 값들을 출력한다. 전가산기는 상기 제2 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 하위 2번째 전가산기로부터 출력되는 합 값을 전가산하여 캐리 입력 값으로 상기 몫 로직으로 제공한다. 캐리전파형 가산기(CPA)는 상기 제2 CSA로부터의 캐리 값들과 합 값들을 가산하여 출력한다.

<24> 바람직하기로, 상기 곱셈 장치는 상기 승수의 비트들을 순차적으로 쉬프트시켜 상기 쉬프트된 비트열을 생성하는 쉬프트 레지스터와, 상기 피승수를 저장하기 위한 레지스터를 더 포함한다.

<25> 바람직하기로, 상기 기록 로직은 상기 생성된 비트열의 하위 2비트들을 부스 기록하는 기록회로와, 상기 부스 기록된 결과와 상기 피승수를 다중화하는 다중화기와, 상기 생성된 비트열의 하위 2비트들에 따라 상기 다중화기의 출력을 선택적으로 1의 보수화하고 상기 $(n+4)$ 비트의 부호있는 이진수들을 출력하는 보수화기를 포함한다.

<26> 바람직하기로, 상기 몫 로직은 상기 캐리 입력값과 상기 제1 CSA의 최하위 전가산기로부터 출력되는 합 값을 전가산하는 전가산기와, 상기 제1 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 2번째 전가산기로부터 출력되는 합 값을 배타적 논리합하는 배타적 논리합기와, 상기 전가산기와 상기 배타적 논리합기의 출력들과 미리 설정된 입력비트를 조합하여 상기 3비트의 결정 값으로 출력하는 조합회로를 포함한다. 또한, 상기 몫 로직은 상기 캐리 입력값을 저

장하기 위한 디플립플롭을 더 포함하고, 상기 먹스 로직의 상기 전가산기 및 상기 배타적 논리합 기로는 미리 설정된 보정용 캐리 값이 제공되고, 상기 먹스 로직의 상기 전가산기로는 상기 피승수의 부호 비트가 제공되고, 상기 먹스 로직의 상기 전가산기의 출력은 상기 제2 CSA의 최하위 전가산기로 제공된다.

<27> 본 발명의 제2 실시예에 따르면, 모듈라 곱셈 연산 장치의 기록 로직은 상기 승수의 비트들이 순차적으로 쉬프트되어 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 상기 피승수를 다중화하여 $(n+3)$비트의 이진 수들을 출력한다. 제1 캐리저장형 가산기(CSA)는 $(n+3)$개의 전가산기들을 구비하고, $(n+1)$비트의 제1 신호와, $(n+2)$비트의 제2 신호와, 상기 기록 로직으로부터의 $(n+43)$비트의 상기 이진 수들을 제3 신호로 입력하고, 상기 제1 신호의 상위 $(n+1)$번째 비트는 상기 전가산기들중 상위 3번째의 전가산기로 입력되고, 상기 제2 신호의 상위 $(n+2)$번째 비트는 상기 전가산기들중 상위 2번째의 전가산기로 입력되고, 상기 전가산기들중 최상위 전가산기로 "0"레벨의 상기 제1 신호 및 상기 제2 신호가 입력되고, 상기 전가산기들중 상위 2번째의 전가산기로 "0"레벨의 상기 제1 신호가 입력되고, 상기 $(n+3)$개의 전가산기들에 의해 $(n+3)$비트의 캐리 값들과 합 값들을 출력한다. 먹스 로직은 상기 제1 CSA로부터의 상기 $(n+3)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들과 하위 1개의 전가산기로부터 출력되는 캐리 값을 입력하고, 모듈라 감소의 배수를 결정하기 위한 2비트의 결정 값을 출력한다. 선택기는 상기 결정 값에 따라 미리 정해진 모듈로 수의 집합중 하나의 모듈로 수를 선택하여 출력한다.

<28> 제2 캐리저장형 가산기(CSA)는

(n+3)개의 전가산기들을 구비하고, 상기 선택기로부터의 선택된 (n+3)비트의 모듈로 수를 제1 신호로서 입력하고, 상기 제1 CSA로부터의 상기 (n+3)비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 (n+2)비트의 캐리 값들을 제2 신호로서 입력하고, 상기 (n+3)비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 (n+2)비트의 합 값들을 제3 신호로서 입력하고, 상기 제1 신호의 (n+3)비트들은 상기 전가산기들의 최하위 전가산기들로부터 순차 입력되고, 상기 제2 신호의 (n+2)비트들은 상기 전가산기들중 하위 2번째 전가산기들로부터 순차 입력되고, 상기 제3 신호의 (n+2)비트들은 상기 전가산기들의 하위 2번째 전가산기들로부터 순차 입력되고, 상기 (n+3)개의 전가산기들에 의해 (n+4)비트의 캐리 값들과 합 값들을 출력한다. 논리곱 연산기는 상기 제2 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 하위 2번째 전가산기로부터 출력되는 합 값을 논리곱 연산하여 캐리 입력 값으로 상기 몫 로직으로 제공한다. 캐리 전파형 가산기(CPA)는 상기 제2 CSA로부터의 캐리 값들과 합 값들을 가산하여 출력한다.

<29> 바람직하기로, 상기 곱셈 장치는 상기 승수의 비트들을 순차적으로 쉬프트시켜 상기 쉬프트된 비트열을 생성하는 쉬프트 레지스터와, 상기 피승수를 저장하기 위한 레지스터를 더 포함한다.

<30> 바람직하기로, 상기 기록 로직은 상기 생성된 비트열의 하위 2비트들에 따라 상기 피승수를 다중화하여 출력하는 다중화기이다.

<31> 바람직하기로, 상기 몫 로직은 상기 캐리 입력값과 상기 제1 CSA의 최하위 전가산기로부터 출력되는 합 값을 반가산하는 반가산기와, 상기 제1 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 2번째 전가산기로부터 출력되는 합 값을 배타적 논리합하는 배타적 논리합기와, 상

도 가능한 한 동일한 참조번호들 및 부호들로 나타내고 있음에 유의해야 한다. 하기에서 본 발명을 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다.

<36> A. 본 발명의 개요

<37> 하기에서 설명될 본 발명은 각각 n비트인 $A=a_{n-1}\cdot 2^{n-1}+\cdots+a_1\cdot 2+a_0$, $B=b_{n-1}\cdot 2^{n-1}+\cdots+b_1\cdot 2+b_0$ $N=n_{n-1}\cdot 2^{n-1}+\cdots+n_1\cdot 2+n_0$ 에 대해서 몽고메리(Montgomery) 알고리즘을 이용해서 모듈라 곱셈 연산 $A\cdot B \bmod N$ 을 수행하기 위한 장치 및 그에 의한 방법을 제안하고 있다. 이때 A는 승수이고, B는 피승수이고, N은 모듈라 수를 나타내며, 이들의 비트 크기는 512 혹은 1024와 같이 큰 수가 될 수 있다.

<38> 상기 모듈라 곱셈 연산 동작 $A\cdot B \bmod N$ 의 구현은 후술되는 2가지 실시예들에 의한다. 각 실시예들은 각각 n 비트인 A,B,N (여기서, $-N\le A,B<N$)을 입력으로 받아서 $m+2$(여기서, $m=n/2$) 클럭 내에 $A\cdot B\cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하는 모듈라 곱셈 장치 및 방법을 제안한다. 이렇게 제안된 모듈라 곱셈 장치에 의한 곱셈 결과를 이용하여 $A\cdot B \bmod N$ 이 계산될 수 있으며,또한 RSA 연산 수행에 필요한 모듈라 멱승 $m^e \bmod N$ 이 될 수 있음이 설명될 것이다. 도면들 도 1 내지 도 6은 본 발명의 제1 실시예에 따른 모듈라 곱셈 장치를 구성하는 요소들의 구성을 보여주는 도면들이고, 도 7 내지 도 13은 본 발명의 제2 실시예에 따른 모듈라 곱셈 장치를 구성하는 요소들의 구성을 보여주는 도면들이다. 그리고 나머지 도 14는 본 발명

기 반가산기와 상기 배타적 논리합기의 출력들과 미리 설정된 입력비트를 조합하여 상기 2비트의 결정 값으로 출력하는 조합회로를 포함한다. 또한, 상기 몫 로직은 상기 캐리 입력값을 저장하기 위한 디플립플롭을 더 포함하고, 상기 몫 로직의 상기 반가산기의 출력은 상기 제2 CSA의 최하위 전가산기로 제공된다.

<32> 이에 따라 본 발명은 암호화/복호화 및 디지털 서명을 생성 및 검증하는데 응용할 수 있다. 즉, 본 발명은 전자적 문서 혹은 데이터를 암호화/복호화하거나 그 문서 혹은 데이터에 서명 기능을 제공할 수 있는 디지털 서명을 생성/검증하는데 이용할 수 있다.

<33> 전술한 바와 같은 내용은 당해 분야 통상의 지식을 가진 자는 후술되는 본 발명의 구체적인 설명으로 보다 잘 이해할 수 있도록 하기 위하여 본 발명의 특징들 및 기술적인 장점들을 다소 넓게 약술한 것이다.

<34> 본 발명의 청구범위의 주제를 형성하는 본 발명의 추가적인 특징들 및 장점들이 후술될 것이다. 당해 분야에서 통상의 지식을 가진 자는 본 발명의 동일한 목적들을 달성하기 위하여 다른 구조들을 변경하거나 설계하는 기초로서 발명의 개시된 개념 및 구체적인 실시예가 용이하게 사용될 수도 있다는 사실을 인식하여야 한다. 당해 분야에서 통상의 지식을 가진 자는 또한 발명과 균등한 구조들이 본 발명의 가장 넓은 형태의 사상 및 범위로부터 벗어나지 않는다는 사실을 인식하여야 한다.

【발명의 구성 및 작용】

<35> 이하 본 발명의 바람직한 실시예의 상세한 설명이 첨부된 도면들을 참조하여 설명될 것이다. 도면들 중 참조번호들 및 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라

도 가능한 한 동일한 참조번호들 및 부호들로 나타내고 있음에 유의해야 한다. 하기에서 본 발명을 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다.

<36> A. 본 발명의 개요

<37> 하기에서 설명될 본 발명은 각각 n비트인 $A=a_{n-1}\cdot 2^{n-1}+\cdots+a_1\cdot 2+a_0$, $B=b_{n-1}\cdot 2^{n-1}+\cdots+b_1\cdot 2+b_0$ $N=n_{n-1}\cdot 2^{n-1}+\cdots+n_1\cdot 2+n_0$ 에 대해서 몽고메리(Montgomery) 알고리즘을 이용해서 모듈라 곱셈 연산 $A\cdot B \bmod N$ 을 수행하기 위한 장치 및 그에 의한 방법을 제안하고 있다. 이때 A는 승수이고, B는 피승수이고, N은 모듈라 수를 나타내며, 이들의 비트 크기는 512 혹은 1024와 같이 큰 수가 될 수 있다.

<38> 상기 모듈라 곱셈 연산 동작 $A\cdot B \bmod N$ 의 구현은 후술되는 2가지 실시예들에 의한다. 각 실시예들은 각각 n 비트인 A,B,N (여기서, $-N\le A,B<N$)을 입력으로 받아서 $m+2$(여기서, $m=n/2$) 클럭 내에 $A\cdot B\cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하는 모듈라 곱셈 장치 및 방법을 제안한다. 이렇게 제안된 모듈라 곱셈 장치에 의한 곱셈 결과를 이용하여 $A\cdot B \bmod N$ 이 계산될 수 있으며,또한 RSA 연산 수행에 필요한 모듈라 멱승 $m^e \bmod N$ 이 될 수 있음이 설명될 것이다. 도면들 도 1 내지 도 6은 본 발명의 제1 실시예에 따른 모듈라 곱셈 장치를 구성하는 요소들의 구성을 보여주는 도면들이고, 도 7 내지 도 13은 본 발명의 제2 실시예에 따른 모듈라 곱셈 장치를 구성하는 요소들의 구성을 보여주는 도면들이다. 그리고 나머지 도 14는 본 발명

의 실시예들에 따른 모듈라 곱셈 장치가 이용될 수 있는 IC 카드의 블록 구성을 보여주는 도면이다.

<39> 후술되는 본 발명의 실시예들에 따른 모듈라 곱셈 장치는 승수의 비트들을 순차적으로 쉬프트시켜 쉬프트된 비트열을 생성하고, 상기 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하는 것을 특징으로 한다. 즉, 대비적으로 종래 기술에 따른 모듈라 곱셈 장치는 승수의 비트들을 순차적으로 쉬프트시키면서 생성되는 하위 1비트만을 기록하였으나, 본 발명에서는 기록의 대상이 하위 2비트가 되도록 처리함으로써 곱셈 연산 동작이 보다 고속으로 수행되도록 한다. 다시 말하면, 본 발명의 실시예에 따른 모듈라 곱셈 장치는 변형된 기록 로직을 구비하고 있으며, 모듈라 곱셈 장치를 구성하는 다른 나머지의 요소들은 상기 변형된 기록 로직에 대응하는 형태로 구성되어 몽고메리 알고리즘에 따른 모듈라 곱셈 연산 동작을 수행한다.

<40> B. 제1 실시예

<41> B-1. 발명의 구성

<42> 도 1은 본 발명의 제1 실시예에 따른 모듈라 곱셈 장치의 구성을 보여주는 도면이다.

<43> 상기 도 1을 참조하면, 상기 모듈라 곱셈 장치는 기록 로직(Recording Logic) 110, 제1 캐리저장형 가산기(Carrier Save Adder)(이하 "CSA"라 칭함) 120, 몫 로직(Quotient Logic) 130, 선택기 140, 제2 CSA 150, 전가산기(FA: Full Adder) 160으로 구성된다. 상기 모듈라 곱셈 장치는 Montgomery 알고리즘에 따라

각각 n 비트인 A,B,N (여기서, $-N \le A,B < N$)을 입력으로 받아서 $m+2$(여기서, $m = n/2$) 클럭 내에 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R = 4^{m+2}$)을 계산하는 모듈라 곱셈 장치에 대한 하드웨어적인 구성이다. 즉, 이 모듈라 곱셈 장치는 $A \cdot B \cdot 2^{-(n+4)} \bmod N$을 계산하기 위한 구성을 갖는다.

<44> 상기 CSA 120,150은 각각 $(n+4)$개 전가산기들이 병렬로 구성되는 것으로, 3비트 입력으로 각각 캐리와 합 비트를 출력한다. 상기 기록 로직 110은 승수 A를 기반으로 변형된 부스 기록(modified Booth recoding) 동작을 수행하며, $(n+4)$ 비트의 부호있는 확장(signed extension) 비트로 0, ± B, ± 2B 값 중 하나를 출력한다. 상기 몫 로직 130은 상기 CSA1 120의 최하위(LSB: Least Significant Bit) 캐리 값 $C_{1,0}$, 합 LSB 2비트 $S_{1,1}, S_{1,0}$ 와 Carry-in, B의 부호 비트를 입력으로 하며, 모듈라 감소(Modular reduction)의 배수를 결정하기 위한 값인 3비트의 $q_2 q_1 q_0$ 을 출력한다. 상기 선택기 140은 상기 결정된 q 값에 따라 $0, \pm 1, \pm 2$중 한 값을 선택하여 출력하기 위한 것으로, 다중화기(MUX: Multiplexer)로 구현이 가능하다. 상기 전가산기 160은 상기 CSA2 150의 출력 2비트인 $S_{2,1}, C_{2,0}$와 캐리값 cin을 입력하여 전가산 동작을 수행하고, 그 전가산 결과값을 캐리-인(Carry-in) 신호로서 상기 몫 로직 130으로 제공한다.

<45> 비록 상기 도 1에서는 도시하고 있지 않지만, 매 클럭마다 상기 CSA2 150의 출력인 캐리 값과 합 값을 각각 저장하기 위한 임시 저장 레지스터(C), 레지스터(R)가 구비되며, 또한 상기 임시 저장 레지스터들(C,R)에 저장된 값들을 가산하여 모듈라 곱셈 결과로서 출력하기 위한 캐리 전파형 가산기(Carry Propagation Adder)가 구비 된다는 사실에 유의할 필요가 있을 것이다.

<46> 도 2는 도 1에 도시된 기록 로직 110의 보다 구체적인 구성을 보여주는 도면이다.

<47> 상기 도 2를 참조하면, 상기 기록 로직 110은 승수(A)의 비트들이 순차적으로 쉬프트되어 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와

피승수(B)를 다중화하여 $(n+4)$비트의 부호있는 이진 수들을 출력한다. 즉, 상기 기록 로직 110의 전단에는 상기 승수의 비트들을 순차적으로 쉬프트시켜 상기 쉬프트된 비트열을 생성하는 쉬프트 레지스터 102와, 상기 피승수를 저장하기 위한 레지스터 104가 구비된다. 상기 기록 로직 110은 기록회로 112와, 다중화기(MUX) 114와, 1의 보수화기 116으로 구성된다. 상기 기록 회로 112는 상기 생성된 비트열의 하위 2비트들(a_{i+1}, a_i)을 부스 기록(Booth recording)한다. 상기 다중화기 114는 상기 부스 기록된 결과(z_{i+1})와 상기 피승수를 다중화하고, 다중화 결과로서 0, B, 2B 를 출력한다. 상기 1의 보수화기 116은 상기 생성된 비트열의 하위 2비트들에 따라 상기 다중화기 114의 출력을 선택적으로 1의 보수화하고 상기 $(n+4)$비트의 부호있는 이진 수들을 출력한다. 전술한 상기 기록 로직 110은 승수 A를 기반으로 변형된 부스 기록 (modified Booth recoding)을 구현한 회로이며, $(n+4)$ 비트의 부호있는 확장(signed extension) 비트를 출력하는데, 이 값은 $0, \pm B, \pm 2B$ 값 중 하나이다.

<48> 도 3은 도 1에 도시된 CSA1 120의 보다 구체적인 구성을 보여주는 도면이다.

<49> 상기 도 3을 참조하면, 상기 CSA1 120은 $(n+4)$개의 전가산기들 121~125를 구비하고, $(n+2)$비트의 제1 신호 $(S_{2,2}\text{-}S_{2,n+3})$와, $(n+3)$비트의 제2 신호 $(C_{2,1}\text{-}C_{2,n+3})$와, 상기 기록 로직 110으로부터의 $(n+4)$비트의 상기 이진 수들을 제3 신호 $B_0 \sim B_{n+3}$로 입력하고, 상기 $(n+4)$개의 전가산기들 121~125에 의해 전가산하여 (n+4)비트의 캐리 값들 $(C_{1,0} \sim C_{1,n+3})$과 합 값들 $(S_{1,0} \sim S_{1,n+3})$을 출력한다. 이때 상기 제1 신호의 상위 $(n+2)$번째 비트$(S_{2,n+3})$는 상기 전가산기들 중 상위 3개의 전가산기들 123~125로 입력되고, 상기 제2 신호의 상위 $(n+3)$번째 비트 $C_{2,n+3}$ 는 상기 전가산기들 중 상위 2개의 전가산기들 124~125로 입력된다.

<50> 도 4는 도 1에 도시된 몫 로직 130의 보다 구체적인 구성을 보여주는 도면이다.

<51> 상기 도 4를 참조하면, 상기 몫 로직 130은 상기 CSA1 120으로부터의 상기 $(n+4)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들$(S_{1,0}, S_{1,1})$과 하위 1개의 전가산기로부터 출력되는 캐리 값$(C_{1,0})$을 입력하고, 모듈라 감소의 배수를 결정하기 위한 3비트의 결정 값$(q_2 q_1 q_0)$을 출력한다. 상기 몫 로직 130은 디(D)플립플롭 132와, 전가산기 134와, 배타적 논리합(XOR)기 136과, 조합회로 138로 구성된다. 상기 디플립플롭 132는 캐리 입력값 Carry-in을 입력하여 일시적으로 저장한다. 상기 전가산기 134는 상기 디플립플롭 132에 저장된 캐리 입력값 Carry-in과 상기 CSA1 120의 최하위 전가산기 121로부터 출력되는 합 값$(S_{1,0})$을 전가산한다. 상기 배타적 논리합기 136은 상기 CSA1 120의 최하위 전가산기 121로부터 출력되는 캐리 값$(C_{1,0})$과 2번째 전가산기 122로부터 출력되는 합 값$(S_{1,1})$을 배타적 논리합한다. 상기 전가산기 134 및 상기 배타적 논리합기 136으로는 미리 설정된 보정용 캐리 값(cin)이 제공되며, 상기 전가산기 134로는 피승수의 부호 비트(B sign)가 또한 제공된다. 상기 조합회로 138은 상기 전가산기 134의 출력(S_0)과 상기 배타적 논리합기 136의 출력(S_1)과 미리 설정된 입력비트(n1)를 조합하여 3비트의 결정 값$(q_2 q_1 q_0)$으로 출력한다.

<52> 도 5는 도 1에 도시된 CSA2 150의 보다 구체적인 구성을 보여주는 도면이다.

<53> 상기 도 5를 참조하면, 상기 CSA2 150은 $(n+4)$개의 전가산기들 151~156을 구비하고 있다. 상기 CSA2 150은 상기 선택기 140으로부터의 선택된 $(n+4)$비트의 모듈로 수$(N: N_0 \sim N_{n+3})$를 제1 신호로서 입력하고, 상기 CSA1 120으로부터의 $(n+4)$비트의 캐리 값들 중 최상위 캐리 값을 제외한 나머지의 $(n+3)$비트의 캐리 값들$(C_{1,0} \sim C_{1,n+2})$을 제2 신호로서 입력하고, 상기 $(n+4)$비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 $(n+3)$비트의 합 값

들($S_{1,1} \sim S_{1,n+3}$)을 제3 신호로서 입력하고, 상기 $(n+4)$개의 전가산기들 151~156에 의해 $(n+4)$비트의 캐리 값들($C_{2,0} \sim C_{2,n+3}$)과 합 값들($S_{2,0} \sim S_{2,n+3}$)을 출력한다. 이때 상기 제1 신호의 $(n+4)$비트들은 상기 전가산기들의 최하위 전가산기들 151로부터 순차 입력되고, 상기 제2 신호의 $(n+3)$비트들은 상기 전가산기들중 하위 2번째 전가산기들 152로부터 순차 입력되고, 상기 제3 신호의 $(n+3)$비트들은 상기 전가산기들의 하위 2번째 전가산기들 152로부터 순차 입력된다. 상기 전가산기들중 최하위 전가산기 151에는 상기 몫 로직 130의 전가산기 134로부터의 출력(S_0)과, $q_{1,2}$ 와, 모듈로 수(N)를 나타내는 최하위 비트(N_0)가 입력된다.

<54> 도 6은 도 1에 도시된 전가산기 160의 보다 구체적인 구성을 보여주는 도면이다.

<55> 상기 도 6을 참조하면, 상기 전가산기 160은 상기 CSA2 150의 최하위 전가산기 151로부터 출력되는 캐리 값($C_{2,0}$)과 하위 2번째 전가산기 152로부터 출력되는 합 값($S_{2,0}$)을 전가산하여 캐리 입력 값(Carry-in)으로 출력한다. 상기 전가산기 160은 전가산 동작을 위해 미리 설정된 보정용 캐리 값(cin)을 또한 제공받으며, 그 전가산 결과로서 캐리 입력 값(Carry-in)을 출력한다. 이 캐리 입력 값(Carry-in)은 상기 몫 로직 130으로 제공된다.

<56> B-2. 발명의 원리

<57> 본 발명은 각각 n비트인 A,B,N (여기서,

$-N \le A,B<N$)을 입력으로 받아서 $m+2$(여기서, $m = n/2$) 클럭 안에 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R = 4^{m+2}$) 을 계산하는 장치에 관한 것이다. 이러한 본 발명의 구현에 사용되는 다음과 같은 3가지의 원리들이 설명될 것이다. 첫째, 모듈라 곱셈 연산을 위한 승수(A) 및 피승수(B)를 표현하는 원리가 설명될 것이다. 둘째, 모듈라 곱셈 연산을 위한 승수(A)를 기록하는 원리가 설명될 것이다. 셋째, 본 발명의 기록 원리를 이용한 Montgomery 알고리즘에 대한 원리가 설명될 것이다.

<58> < Number Representation >

<59> 본 발명에서는 모듈라 곱셈 연산을 위하여 승수(A)와 피승수(B)를 부호있는 이진 수로 표현(Signed binary representation)한다. 즉, 각각 n비트인 A, B 는 부호있는 동작(signed operation)을 위해서 각각 $(n+4)$비트로 변환되는데, 이러한 변환 과정중에 값이 음수인 경우는 1의 보수 형태로 변환된다.

<60> < Booth's Recording >

<61> 본 발명에서는 본 발명이 출원되기 이전에 당해 분야 통상의 지식을 가진 자에게 잘 알려진 부스 기록(Booth Recording) 방식을 변형하여 변형된 부스 기록(Modified Booth Recording) 방식을 사용하는 것을 특징으로 한다. 이러한 본 발명의 특징은 모듈라 곱셈 연산이 보다 빠른 속도로 수행될 수 있도록 하는 효과를 가진다. 승수 A는 상기 변형된 부스 기록

방식에 의해 2 비트씩 z_i(여기서, $0 \le i \le m+1$)로 기록된다. 이때 $a_{m+4} = a_{m+3} = a_{-1} = 0$로 가정한다. 하기의 <표 1>은 본 발명에 의한 변형된 부스 기록의 규칙을 보여주고 있다.

<62> 【표 1】

a_{i+1}	a_i	a_{i-1}	z_{i+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

<63> < Booth's Recoding을 이용한 Radix-4 Montgomery Algorithm >

<64> 다음의 알고리즘은 본 발명이 radix-4 Montgomery 모듈라 곱셈을 위해서 상기 변형된 부스 기록 방식을 이용하고 있음을 보여준다. 원래 Montgomery 알고리즘에서는 결과 값과 modulus N과 비교하여 결과 값이 modulus N보다 크면 한번 뺄셈 연산을 해야 하나, 하기에서 보여지는 알고리즘에서는 그 절차가 없음을 알 수 있다.

<65>

Input: N, $-N \le A, B < N$

Output: $S = A \cdot B \cdot 4^{-m-2} \bmod N$, $-N < S < N$

$S \leftarrow 0$ (1)

$for\ i = 0\ to \left\lceil \frac{n+1}{2} \right\rceil$ (2)

$S = S + A_i \times B$ (3)

$q_{i(2,1,0)} = f(s_1, s_0, n_1, n_0)$ (4)

$S = S + q_i \times N$ (5)

$S = S / 2^2$ (6)

$endfor$ (7)

【수학식 1】

<66> 상기 <표 2>에 나타낸 알고리즘에서, (3)번 절차의 A_i는 부스 기록된 두 비트를 의미하며 $-2 < A_i < 2$ 의 값을 갖는다. (4)번 절차는 (5)번 절차의 결과 값의 LSB(Least Significant Bit) 두 비트가 '0' 이 되도록 하는 함수를 의미한다. 상기 (4)번 절차에서의 결과 값은 입력비트 s_1, s_0, n_1, n_0에 의존하며 다음의 <표 4>에서 보인 바와 같이 결정된다. 모듈라 감소(modular reduction)에 사용되는 값 q_i의 MSB(Most Significant Bit)인 q_{i2}는 부호 비트이며, q_i는 집합 {0, ±, 2}원소들 중 하나이다. q_i는 다음의 <표 3>에 의해서 계산된다.

<67> 【수학식 2】 $q_0 = s_0$

<68> $q_1 = s_0 s_1$

<69> $q_2 = s_0 s_1 n_1 + s_0 s_1 n_1$

<70> 【표 2】

s_0	s_1	n_1	q_2	$q_1 q_0$
0	0	0	0	00
0	0	1	0	00
0	1	0	0	10
0	1	1	0	10
1	0	0	1	01
1	0	1	0	01
1	1	0	0	01
1	1	1	1	01

<71> B-3. 발명의 동작

<72> 본 발명은 각각 n비트인 A,B,N(여기서, $-N \le A,B < N$)을 입력으로 받아서 $m+2$(여기서, $m=n/2$) 클럭 안에 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R = 4^{m+2}$)을 계산하기 위한 것으로, 도 1에 도시된 바와 같은 장치에 의해 그 계산 동작이 수행된다.

<73> 먼저, 도 1에 도시된 장치에 의해 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하는 과정이 설명될 것이다. 하기의 a) 단계는 초기화 단계이고, b)~h) 단계들은 매 클럭마다 수행되는 단계들이고, i) 단계는 상기 b)~h) 단계들을 $(m+2)$ 클럭을 수행한 후에 수행되는 단계이다.

<74> a) 모듈라 곱셈 연산을 위해 입력되는 n비트인 A,B,N이 각각의 레지스터(혹은 메모리)들에 저장된다. 본 발명의 장치에 따르면, 입력들 A, B는 도 2에 도시된 레지스터들 102, 104에 저장되는 것으로 도시하고 있고, N을 저장하는 별도의 레지스터에 대해서는 도시하고 있지 않지만 이러한 레지스터가 사용될 수 있다는 사실에 유의하여야 한다. 이때 A를 저장하기 위한 레지스

터 102는 매 클럭마다 오른쪽으로 2비트 단위로 이동시키는 쉬프트 레지스터이다. 편의상 A를 저장하기 위한 레지스터는 레지스터 A로, B를 저장하기 위한 레지스터는 레지스터 B로 설명될 수 있을 것이다. 만약, 메모리인 경우는 한 워드 단위로 값을 읽어내게 된다. 도 1의 CSA2 150에 의한 계산 결과를 임시적으로 저장하기 위한 임시 레지스터들(혹은 메모리) C와 S(도시하지 않음)가 0으로 초기화된다.

<75> b) 각각의 레지스터들 102,104에 모든 데이터들이 입력되었을 때, 기록 로직 110의 부스 기록 회로 112는 레지스터 102의 LSB 두 비트를 기반으로 부스 기록 기능을 수행한다. 상기 기록 로직 110의 MUX 114는 레지스터 104에 저장된 B 값을 입력하고, 레지스터 102의 LSB 두 비트에 따라서 $0, \pm B, \pm 2B$ 값 중 하나를 생성해서 CSA1 120의 3개의 입력 중 하나의 입력으로서 제공한다. 이때 상기 기록 로직 110의 1의 보수화기 116은 레지스터 102의 LSB 두 비트에 따라서 $0, \pm B, \pm 2B$ 값 중 한 값을 1의 보수로 바꾼 후 $n+4$비트 수로 나타내어 CSA1 120의 3개의 입력 중 하나의 입력으로서 제공된다.

<76> c) CSA1 120은 $n+4$비트의 부호있는 2진수(binary signed number) 3개를 입력으로 받아서 덧셈 동작을 수행한다. 상기 CSA1 120은 $n+4$개의 전가산기들 121~125로 구성된다. 이때 이전 단계의 전가산기에서 발생되는 캐리는 다음 단계의 전가산기로 제공되지만, MSB의 전가산기 125에서 발생되는 캐리는 무시된다.

<77> d) 묶 로직 130은 상기 CSA1 120의 출력 값들 $S_{1,1}, C_{1,0}, S_{1,0}$ 과, 전가산기 160으로부터 제공되는 Carry-in 신호와, 피승수 B의 부호 비트(B sign)를 입력하고, 전가산기 134와 배타적 논리합기 136에 의해 각각

S_1, S_0을 계산하여 출력한다. 이때 상기 전가산기 134 및 배타적 논리합기 136으로는 일종의 보정용 캐리신호인 cin이 입력된다. 이 캐리신호 cin은 2의 보수를 사용하는 기존의 Booth 기록 방식과 달리 본 발명이 1의 보수를 사용함에 따라 양자의 차이를 보정하기 위한 신호이다.

<78> e) 상기 룩 로직 130의 조합 회로 138은 상기 d) 단계에서 계산된 S_1, S_0을 입력하고, 상기 <표 4>에 기재한 바와 같은 진리표에 의해서 3비트 값인 q 값을 결정한다. 비록 상기 <표 4>에 기재한 진리표에 의해 q값을 결정하는 회로에 대한 구체적인 구성을 도시하고 있지는 않으나, 당해 분야 통상의 지식을 가진 자라면 이러한 결과값을 결정하는 회로를 일반적인 논리 조합회로에 의해 용이하게 구현 할 수 있을 것이다.

<79> f) CSA2 150은 상기 c) 단계에서 구한 CSA1 120의 출력인 캐리와 합 값들, 상기 e) 단계에서 구한 q값들의 LSB 두 비트에 의해 결정된 $0, \pm N, 2N$값 중 한 값을 선택해서 $n+4$ 비트의 부호있는 이진수(binary signed number)를 입력으로 해서 $n+4$ 비트 부호있는 연산을 수행한다. 상기 CSA2 150은 상기 CSA1 120과 마찬가지로 $n+4$개의 전가산기들 151~156으로 구성된다. 이때 상기 전가산기들 151~156은 최하위의 전가산기인 LSB의 전가산기 151의 캐리 입력으로 상기 e) 단계에서 계산된 q값의 MSB 값 $q_{1,2}$가 입력된다.

<80> g) 전가산기 160은 CSA2 150의 출력 값들중 $S_{2,1}$, $C_{2,0}$ 비트들과 보정용 캐리신호인 cin 비트를 입력하고 전가산하여 Carry-in 비트를 출력한다. 이러한 전가산 동작은 전술한 바와 같이 2의 보수를 사용해야 하는 기존의 Booth 기록 방식과 달리 1의 보수를 사용하는 본 발명과의 결과적인 보정을 위한 것이다.

<81> h) 상기 CSA2 150의 출력중 MSB로부터의

$(m+2)$ 개의 합 값들과 항 $(m+3)$ 개의 캐리 값들이 CSA1 120의 입력으로 피드백된다. 이때 상기 CSA2 150의 최상위 가산기 156으로부터 출력되는 합 값의 MSB인 $S_{2,m+3}$을 복사하여 앞에 두 비트 추가하며, 캐리 값의 MSB인 $C_{2,m+3}$을 복사하여 한 비트 추가하여 CSA1 120의 입력 값으로 한다. 즉, 상기 CSA2 150의 가산기 156으로부터 출력되는 합 값 $S_{2,m+3}$은 CSA1 120의 최상위로부터 3개의 가산기들 123~125로 제공되고, 캐리 값 $C_{2,m+3}$은 CSA1 120의 최상위로부터 2개의 가산기들 124,125로 제공된다.

<82> i) (m+2) 클럭 동안에 상기 b)~h) 단계들이 수행된 이후에는 다음과 같은 동작이 수행된다. 즉, CPA(Carry Propagation Adder)(도시하지 않음)는 상기 CSA2 150의 출력인 캐리 값과 합 값에 대해 가산 동작을 수행한다. 이때 가산 결과 값이 음수이면 modulus N을 더하며, 양수이면 modulus N을 더하지 않는다.

<83> 예를 들어, A, B, N이 각각 다음의 <표 5>와 같이 12비트일 때 상기 절차에 의한 Montgomery 모듈러 연산 결과는 <표 6> 및 <표 7>에 나타낸 바와 같다.

<84>

N=0000.1010.0101.1001 (0xA59) B=0000.0101.1100.0011 (0x5C3)

N'=1111.0101.1010.0110 B'=1111.1010.0011.1100

2N=0001.0100.1011.0010 2B'=1111.0100.0111.1001

【수학식 3】 A=0000.1001.0011.1110 (0x93E)

<85>

【표 3】

i	A_i	CSA 1-out S C	B-sign	Carry in	$S_1 S_0$	C
i	0	0000.0000.0000.0000 0.0000.0000.0000.000	0	0	00	0
0	-2	1111.0100.0111.1001 0.0000.0000.0000.000	1	0	10	1
1	0	1111.0010.0010.1010 0.0001.0000.0010.100	0	1	11	0
2	0	1111.0011.0000.0000 0.0001.0000.0010.100	0	1	01	0
3	1	1111.1000.1111.0000 0.0000.1011.0000.011	0	1	11	0
4	1	1111.1110.1000.0000 0.0000.1010.1101.011	0	1	11	0
5	-2	0000.1110.1001.0010 1.1110.1010.1101.001	1	1	10	1
6	1	1111.1110.1011.0110 0.0000.1010.1001.001	0	1	01	0
7	0	1111.1111.0011.1011 0.0000.0000.0000.000	0	1	00	1

<86> 【표 4】

i	A_i	$S_1 S_0$	C	$q_2 q_1 q_0$	CSA 2 out S C	Carry in
i	0	00	0	000	0000.0000.0000.0000 0.0000.0000.0000.000	0
0	-2	10	1	010	(11).1110.0000.1100.1010 (0)0.0010.1000.0110.000	1
1	0	11	0	001	(11).1110.1000.0101.0010 (0)0.0010.0100.0101.001	1
2	0	01	0	101	(00).0001.0110.1000.1110 (1)1.1110.0010.0100.001	1
3	1	11	0	001	(11).1111.1001.1010.1110 (0)0.0001.0100.1010.001	1
4	1	11	0	001	(11).1111.1110.0000.1110 (0)0.0001.0101.1010.001	1
5	-2	10	1	010	(11).1111.0000.1111.0010 (0)0.0001.1101.0010.010	1
6	1	01	0	101	(00).0000.0001.1000.0010 (1)1.1111.1101.0110.111	1
7	0	00	1	000	1111.1111.1011.1010 0.0000.0000.0000.000	1

<87> 다음에, 전술한 바와 같은 본 발명의 장치에 의한 연산 결과 값을 이용하여 모듈라 곱셈 연산 $A \cdot B \bmod N$을 계산하는 절차를 설명하기로 한다. 이에 대한 구체적인 하드웨어 구성이 생략되고 있음에 유의하여야 한다.

<88> 1) 미리 $P = 2^{2(n+4)} \bmod N$을 계산한다.

<89> 2) 본 발명의 장치를 이용하여 $C = A \cdot B \cdot 2^{-(n+4)} \bmod N$을 계산한다.

<90> 3) $P \cdot C \cdot 2^{-(n+4)} \bmod N = A \cdot B \bmod N$을 계산한다.

<91> 그 다음에, 전술한 바와 같은 본 발명의 장치에 의한 연산 결과 값을 이용하여 RSA 연산 수행에 필요한 모듈라 멱승 $m^e \bmod N$을 계산하는 절차를 설명하기로 한다.

<92> 1) 지수 e를 레지스터(혹은 메모리)에 저장한다.

<93> 2) 레지스터 C에 modulus N을 저장한다.

<94> 3) 임시 레지스터 C와 S를 0으로 초기화한다.

<95> 4) Montgomery 모듈라 곱셈 $m' = f_m(m,P,N) = m \cdot P \cdot R^{-1} \bmod N$을 수행한다. 단, 멱승 연산의 밑 P는 위 절차에서 정의한 미리 계산된 값을 나타내며, $R = 2^{n+4}$이다.

<96> 5) m'를 레지스터 B에 로딩한다.

<97> 6) 레지스터 B에 로딩한 값을 이용해서 모듈라 제곱 연산을 수행한다. 이때 Montgomery 모듈라 곱셈에 필요한 승수 A는 레지스터 B에서 로딩하며, 변형된 부스 기록 회로를 이용해서 값을 얻는다.

<98> 7) 지수 e를 왼쪽으로 쉬프트 한다.

<99> 8) 지수 e의 MSB(Most Significant Bit) 1을 무시하고 다음 비트부터 다음 9) - 10) 단계를 수행한다.

<100> 9) 지수 e의 매 비트가 0 또는 1 에 상관없이 단계 4) -5)를 수행하여 모듈라 제곱 연산을 수행한다. 이때 제곱 연산에 필요한 승수는 레지스터 A, 피승수는 레지스터 B에 저장된다.

<101> 10) 지수 e의 현재 비트가 1인 경우에는 단계 9)을 수행한 후, 단계 4) -5)를 수행하여 모듈라 곱셈 연산을 수행한다. 이때 피승수는 레지스터 B 의 내용이고 승수는 멱승 연산의 밑 m'이다

<102> 11) 지수 e의 모든 비트에 대해서 단계 8)- 10)을 수행한 후 단계 4)를 이용해서 모듈라 곱셈을 한 번 더 수행한다. 이때 피승수는 레지스터 B의 내용이고 승수는 1이다.

<103> 상기 단계들 1)- 11)을 수행한 후 레지스터 C와 S에 남아 있는 값에 대해 CPA(Carry Propagation Adder)를 수행한 값이 음수이면 modulus N을 더하며, 양수이면 modulus N을 더하지 않는 값이 최종적인 멱승 값 $m^e \bmod N$이 된다.

<104> B-4. 발명의 효과

<105> 전술한 바와 같이, 본 발명은 $A \cdot B \cdot 2^{-(n+4)} \bmod N$을 계산하는 회로를 기술한 것으로 기술된 회로를 이용하여 상기 일반적인 모듈라 곱셈인 $A \cdot B \bmod N$을 계산할 수 있다. 이러한 본 발명에 기초하여 계산된 $A \cdot B \bmod N$는 디지털 서명을 생성 및 검증하기 위한 기기에 사용될 수 있는 하드웨어적인 장치에 사용될 수 있다. 또한, 본 발명은 IC 카드를 바탕으로 한 전자 서명, 인증, 암/복호화의 하드웨어 장치에 사용될 수 있다. 또한, 본 발명은 모듈라 곱셈을 수행하는 전자

서명 기기에 의해서 암호 및 복호화 하는 기기를 제공할 수 있다. 또한, 본 발명은 전자 서명 기기를 바탕으로 해서, NIST-DSS, RSA, ElGamal, Schnorr 전자 서명 등 기존의 공개키 암호 시스템을 구현하는데 이용될 수 있다.

<106> C. 제2 실시예

<107> C-1. 발명의 구성

<108> 도 7은 본 발명의 제2 실시예에 따른 모듈라 곱셈 장치의 구성을 보여주는 도면이다.

<109> 상기 도 7을 참조하면, 상기 모듈라 곱셈 장치는 기록 로직(Recording Logic) 210, 제1 캐리저장형 가산기(Carrier Save Adder)(이하 "CSA"라 칭함) 220, 몫 로직(Quotient Logic) 230, 선택기 240, 제2 CSA 250, 논리곱 연산기(AND) 260으로 구성된다. 상기 모듈라 곱셈 장치는 Montgomery 알고리즘에 따라 각각 n 비트인 A,B,N (여기서, $-N \le A,B < N$)을 입력으로 받아서 $m+2$(여기서, $m=n/2$) 클럭 내에 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하는 모듈라 곱셈 장치에 대한 하드웨어적인 구성이다. 즉, 이 모듈라 곱셈 장치는 $A \cdot B \cdot 2^{-(n+4)} \bmod N$을 계산하기 위한 구성을 갖는다.

<110> 상기 CSA 220,250은 각각

$(n+4)$개 전가산기들이 병렬로 구성되는 것으로, 3비트 입력으로 각각 캐리와 합 비트를 출력한다. 상기 기록 로직 210은 승수 A를 기반으로 변형된 부스 기록(modified Booth recoding) 동작을 수행하며, $(n+3)$ 비트의 0,B,2B,3B 값 중 하나를 선택하여 출력한다. 상기 몫 로직 230은 상기 CSA1 220의 최하위(LSB: Least Significant Bit) 캐리 값 $C_{1,0}$, 합 LSB 2비트 $S_{1,1}, S_{1,0}$ 와 Carry-in, B의 부호 비트를 입력으로 하며, 모듈라 감소(Modular reduction)의 배수를 결정하기 위한 값인 2비트의 $q_1 q_0$ 을 출력한다. 상기 선택기 240은 상기 결정된 q 값에 따라 0,N,2N,3N 중 한 값을 선택하여 출력하기 위한 것으로, 다중화기(MUX: Multiplexer)로 구현이 가능하다. 상기 논리곱 연산기 260은 상기 CSA2 250의 출력 2비트인 $S_{2,1}, C_{2,0}$을 입력하여 논리곱 연산 동작을 수행하고, 그 연산 결과값을 Carry-in 신호로서 상기 몫 로직 230으로 제공한다.

<111> 비록 상기 도 7에서는 도시하고 있지 않지만, 매 클럭마다 상기 CSA2 250의 출력인 캐리 값과 합 값을 각각 저장하기 위한 임시 저장 레지스터(C), 레지스터(R)가 구비되며, 또한 상기 임시 저장 레지스터들(C,R)에 저장된 값들을 가산하여 모듈라 곱셈 결과로서 출력하기 위한 캐리 전파형 가산기(Carry Propagation Adder)가 구비 된다는 사실에 유의할 필요가 있을 것이다.

<112> 도 8은 도 7에 도시된 기록 로직 210의 보다 구체적인 구성을 보여주는 도면이다.

<113> 상기 도 8을 참조하면, 상기 기록 로직 210은 승수(A)의 비트들이 순차적으로 쉬프트되어 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 피승수(B)를 다중화하여 $(n+3)$비트의 이진 수들을 출력한다. 즉, 상기 기록 로직 210의 전단에는 상기 승수의 비트들을 순차적으로 쉬프트시켜 상기 쉬프트된 비트열을 생성하는 쉬프트 레지스터 202와, 상기 피승수를 저장하기 위한 레지스터 204가 구비된다. 상기 기록 로직 210

은 다중화기(MUX) 212로 구성된다. 상기 다중화기 212는 상기 생성된 비트열의 하위 2비트들(a_{i+1}, a_i)과 상기 피승수를 다중화하고, 다중화 결과로서 0,B,2B,3B를 출력한다. 전술한 상기 기록 로직 210은 승수 A를 기반으로 변형된 부스 기록(modified Booth recoding)을 구현한 회로이며, $(n+3)$ 비트의 0,B,2B,3B 값 중 하나를 선택하여 출력한다.

<114> 도 9는 도 7에 도시된 CSA1 220의 보다 구체적인 구성을 보여주는 도면이다.

<115> 상기 도 9를 참조하면, 상기 CSA1 220은 $(n+4)$개의 전가산기들 221~225를 구비하고, $(n+1)$비트의 제1 신호 $(S_{2,2} \sim S_{2,n+2})$와, $(n+2)$비트의 제2 신호 $(C_{2,1} \sim C_{2,n+2})$와, 상기 기록 로직 210으로부터의 $(n+3)$비트의 상기 이진 수들을 제3 신호 $(B_0 \sim B_{n+2})$로 입력하고, 상기 $(n+3)$개의 전가산기들 221~225에 의해 전가산하여 $(n+3)$비트의 캐리 값들 $(C_{1,0} \sim C_{1,n+3})$과 합 값들$(S_{1,0} \sim S_{1,n+3})$을 출력한다. 상기 제1 신호 및 제2 신호는 상기 CSA2 250으로부터 제공되는 신호이고, 상기 제3 신호는 상기 기록 로직 210으로부터 제공되는 신호이다. 이때 상기 제1 신호의 최상위 비트$(S_{2,n+3})$는 상기 전가산기들중 상위 3번째 전가산기 223으로 입력되고, 상기 제2 신호의 최상위 비트 $C_{2,n+3}$는 상기 전가산기들중 상위 2번째 전가산기 224로 입력된다. 상기 전가산기들중 최상위 전가산기 225에는 제1 신호 및 제2 신호로서 "0"이 제공되고, 2번째 전가산기 224에는 제1 신호로서 "0"이 제공된다. 즉, 상기 CSA1 220을 구성하는 최하위의 전가산기 221로부터 $(n+1)$번째 전가산기 223까지 $(n+1)$비트의 제1 신호 S가 순차 입력되고, $(n+2)$번째 전가산기 224 및 $(n+3)$번째 전가산기 225에는 "0"의 제1 신호가 입력된다. 또한, 상기 CSA1 220을 구성하는 최하위의 전가산기 221로부터 $(n+2)$번째 전가산기 224까지 $(n+2)$비트의 제2 신호$(C_{2,1} \sim C_{2,n+2})$가 순차 입력되고, 번째 전가산기 225에는 "0"의 제2 신호가 입력된다. 또한, 상기 CSA1 220을 구성하는 최하위의 전가산기 221로부터 $(n+1)$번째 전가산기 223까지 $(n+3)$비트의 제3 신호$(B_0 \sim B_{n+2})$가 순차 입력된다.

<116> 도 10은 도 7에 도시된 몫 로직 230의 보다 구체적인 구성을 보여주는 도면이다.

<117> 상기 도 10을 참조하면, 상기 몫 로직 230은 상기 CSA1 220으로부터의 상기 $(n+3)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들$(S_{1,0}, S_{1,1})$과 하위 1개의 전가산기로부터 출력되는 캐리 값$(C_{1,0})$을 입력하고, 모듈라 감소의 배수를 결정하기 위한 2비트의 결정 값$(q_1 q_0)$을 출력한다. 상기 몫 로직 230은 디(D)플립플롭 232와, 반가산기(HA: Half Adder) 234와, 배타적 논리합(XOR)기 236과, 조합회로 238로 구성된다. 상기 디플립플롭 232는 논리곱(AND) 연산기 260으로부터의 캐리 입력값 Carry-in을 입력하여 일시적으로 저장한다. 상기 반가산기 234는 상기 디플립플롭 232에 저장된 캐리 입력값 Carry-in과 상기 CSA1 220의 최하위 전가산기 221로부터 출력되는 합 값($S_{1,0}$)을 반가산한다. 상기 배타적 논리합기 236은 상기 CSA1 220의 최하위 전가산기 221로부터 출력되는 캐리 값($C_{1,0}$)과 2번째 전가산기 222로부터 출력되는 합 값($S_{1,1}$)을 배타적 논리합한다. 상기 조합회로 238은 상기 반가산기 134의 출력(S_0)과 상기 배타적 논리합기 236의 출력(S_1)과 미리 설정된 입력비트(n1)를 조합하여 2비트의 결정 값($q_1 q_0$)으로 출력한다.

<118> 도 11은 도 7에 도시된 CSA2 250의 보다 구체적인 구성을 보여주는 도면이다.

<119> 상기 도 11을 참조하면, 상기 CSA2 250은 $(n+3)$개의 전가산기들 251~256을 구비하고 있다. 상기 CSA2 250은 상기 선택기 240으로부터의 선택된 $(n+3)$비트의 모듈로 수(N: $N_0 \sim N_{n+3}$)를 제1 신호로서 입력하고, 상기 CSA1 220으로부터의 $(n+3)$비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 $(n+2)$비트의 캐리 값들($C_{1,0} \sim C_{1,n+2}$)을 제2 신호로서 입력하고, 상기 $(n+3)$비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 $(n+2)$비트의 합 값들($S_{1,1} \sim S_{1,n+3}$)을 제3 신호로서 입력하고, 상기 $(n+3)$개의 전가산기들 251~256에 의해 $(n+3)$비트의 캐리 값

들($C_{2,0} \sim C_{2,n+2}$)과 합 값들($S_{2,0} \sim S_{2,n+2}$)을 출력한다. 이때 상기 제1 신호의 ($n+3$)비트들은 상기 전가산기들의 최하위 전가산기들 251로부터 순차 입력되고, 상기 제2 신호의 ($n+2$)비트들은 상기 전가산기들중 하위 2번째 전가산기들 252로부터 순차 입력되고, 상기 제3 신호의 ($n+2$)비트들은 상기 전가산기들의 하위 2번째 전가산기들 252로부터 순차 입력된다. 상기 전가산기들중 최하위 전가산기 251에는 상기 몫 로직 230의 반가산기 234로부터의 출력(S_0)과, 논리곱(AND) 연산기 260으로부터의 캐리 입력값 Carry-in이 입력된다.

<120> 도 12는 도 7에 도시된 논리곱 연산기 260의 보다 구체적인 구성을 보여주는 도면이다.

<121> 상기 도 12를 참조하면, 상기 논리곱 연산기 260은 상기 CSA2 250의 최하위 전가산기 251로부터 출력되는 캐리 값($C_{2,0}$)과 하위 2번째 전가산기 252로부터 출력되는 합 값($S_{2,1}$)을 전가산하여 캐리 입력 값(Carry-in)으로 출력한다. 상기 캐리 입력 값(Carry-in)은 상기 몫 로직 230으로 제공된다.

<122> C-2. 발명의 원리

<123> 본 발명은 각각 n비트인 A,B,N (여기서, $-N \le A,B < N$)을 입력으로 받아서 $m+2$ (여기서, $m=n/2$) 클럭 안에 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$) 을 계산하는 장치에 관한 것이다. 이러한 본 발명의 구현에 사용되는 다음과 같은 3가지의 원리들이 설명될 것이다. 첫째, 모듈라 곱셈 연산을 위한 승수(A) 및 피승수(B)를 표현하는 원리가 설명될 것이다. 둘째, 본 발명의 기록 원리를 이용한 Montgomery 알고리즘에 대한 원리가 설명될 것이다.

<124> < 2bit scanning >

<125> 본 발명에서는 승수(A)를 각 클럭에 LSB부터 두 비트씩 스캐닝(scanning)(혹은 쉬프팅(shifting))하여 피승수(B)와 곱하고, 이러한 곱셈 결과를 Montgomery 알고리즘에 이용한다. 그러므로 매 루프에서 발생되는 "a_i"는 {0, 1, 2, 3}의 원소 중 하나이며 이를 피승수(B)와 곱하여 CSA1 220의 입력으로 이용한다.

<126> < Radix-4 Montgomery algorithm >

<127> 다음의 <표 8>에 나타낸 바와 같은 알고리즘은 본 발명이 radix-4 Montgomery 모듈라 곱셈을 이용함을 보여준다. 원래 Montgomery 알고리즘에서는 결과 값과 modulus N과 비교하여 결과 값이 modulus N보다 크면 한번 뺄셈 연산을 해야 하나, 하기에서 보여지는 알고리즘에서는 그 절차가 없음을 알 수 있다.

<128>

【표 5】

Input: N, $0 \le A, B < N$

Output: $S = A \cdot B \cdot 4^{-m-2} \bmod N$, $0 < S < N$

$S = 0$ (1)

$for\ i = 0\ to \left\lceil \frac{n+1}{2} \right\rceil$ (2)

$S = S + A_i \times B$ (3)

$q_{i+1,0} = f(s_1, s_0, n_1, n_0)$ (4)

$S = S + q_i \times N$ (5)

$S = S/2^2$ (6)

$endfor$ (7)

<129> 상기 <표 8>에 나타낸 알고리즘에서 (3)번 절차의 A_i는 A를 두 비트씩 스캔한 값을 의미한다. (4)번 절차는 (5)번 절차의 결과 값의 LSB(Least Significant Bit) 두 비트가 '0'이 되도록 하는 함수를 의미한다. 상기 (4)번 절차에서의 결과 값은 입력비트 s_1, s_0, n_1, n_0에 의존하는데, Montgomery modular 곱셈을 위해서는 N이 홀수이므로 이때 n_0는 항상 1이므로 실제 다음의 <표 10>에서 보인 바와 같이 결정된다. 모듈라 감소(modular reduction)에 사용되는 값 q_i는 집합 {0, 1, 2, 3}원소들 중 하나이다. q_i는 다음의 <표 9>에 의해서 계산된다.

<130> $q_0 = s_0$

【수학식 4】 $q_1 = s_0 \overline{s_1 n_1} + s_0 s_1 + s_1 n_1$

<131> 【표 6】

s_0	s_1	n_1	q_1	q_0
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

<132> C-3. 발명의 동작

<133> 본 발명은 각각 n비트인 A,B,N(여기서, $0 \le A,B < N$)을 입력으로 받아서 $m+2$ (여기서, $m = n/2$) 클럭 안에 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R = 4^{m+2}$)을 계산하기 위한 것으로, 도 7에 도시된 바와 같은 장치에 의해 그 계산 동작이 수행된다.

<134> 먼저, 도 7에 도시된 장치에 의해 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R = 4^{m+2}$)을 계산하는 과정이 설명될 것이다. 하기의 a) 단계는 초기화 단계이고, b)~h) 단계들은 매 클럭마다 수행되는 단계들이고, i) 단계는 상기 b)~h) 단계들을 ($m+2$) 클럭 수행한 후에 수행되는 단계이다.

<135> a) 모듈라 곱셈 연산을 위해 입력되는 n비트인 A, B, N이 각각의 레지스터(혹은 메모리)들에 저장된다. 또한, $n+2$비트인 2B,3B가 각각의 레지스터(혹은 메모리)에 저장된다. 본 발명의 장치에 따르면, 입력되는 비트들 A, B는 도 8에 도시된 레지스터들 202,204에 저장되는 것으로 도시하고 있고, 2B 및 3B를 저장하는 별도의 레지스터에 대해서는 도시하고 있지 않지만 이러

한 레지스터가 사용될 수 있다는 사실에 유의하여야 한다. 이때 A를 저장하기 위한 레지스터 202는 매 클럭마다 오른쪽으로 2비트 단위로 이동시키는 쉬프트 레지스터이다. 편의상 A를 저장하기 위한 레지스터는 레지스터 A로, B를 저장하기 위한 레지스터는 레지스터 B로 설명될 수 있을 것이다. 만약, 메모리인 경우는 한 워드 단위로 값을 읽어내게 된다. 도 7의 CSA2 150에 의한 계산 결과를 임시적으로 저장하기 위한 임시 레지스터들(혹은 메모리) C와 S(도시하지 않음)가 0으로 초기화된다.

<136> b) 각각의 레지스터들 202,204에 모든 데이터들이 입력되었을 때, 기록 로직 110에서는 A 레지스터 202의 LSB 두 비트를 기반으로 부스 기록 기능을 수행한다. 상기 기록 로직 110의 MUX 212는 B 레지스터 204의 값을 입력하고, A 레지스터 202의 LSB 두 비트에 따라서 0,B,2B,3B 값 중 하나를 선택해서 CSA1 220의 3개의 입력 중 하나의 입력으로서 제공한다.

<137> c) CSA1 220은 $n+3$ 비트 2진 수(binary unsigned number) 3개를 입력으로 받아서 덧셈 동작을 수행한다. CSA1 220은 $n+3$개의 전가산기들 221~225로 구성된다.

<138> d) 몫 로직 230은 상기 CSA1 220의 출력 값들 $S_{1,1}$, $C_{1,0}$, $S_{1,0}$과, 논리곱 연산기 260으로부터 제공되는 Carry-in 신호를 입력하고, 반가산기 234와 배타적 논리합기 236에 의해 각각 S_1, S_0을 계산하여 출력한다.

<139> e) 상기 몫 로직 230의 조합 회로 238은 상기 d) 단계에서 계산된 S_1, S_0을 입력하고, 상기 <표 10>에 기재한 바와 같은 진리표에 의해서 2비트 값인 q 값을 결정한다. 비록 상기 <표 10>에 기재한 진리표에 의해 q값을 결정하는 회로에 대한 구체적인 구성을 도시하고 있지는 않으나, 당해 분야 통상의 지식을 가진 자라면 이러한 결과값을 결정하는 회로를 일반적인 논리 조합회로에 의해 용이하게 구현 할 수 있을 것이다.

<140> f) CSA2 250은 상기 c) 단계에서 구한 CSA1 120의 출력인 캐리와 합 값들, 상기 e) 단계에서 구한 q값들의 LSB 두 비트에 의해 결정된 0,N,2N,3N 값 중 한 값을 선택하고 $n+3$ 비트의 이진수를 입력으로 해서 $n+3$ 비트 부호없는 연산을 수행한다. 상기 CSA2 250은 상기 CSA1 220과 마찬가지로 $n+3$개의 전가산기들 251~256으로 구성된다. 이때 상기 전가산기들 251~256은 최하위의 전가산기인 LSB의 전가산기 251의 캐리 입력으로 전단의 Carry_in 신호가 입력됨에 유의하여야 한다.

<141> g) 논리곱(AND) 연산기 260은 CSA2 250의 출력 값들중 $S_{2,1}$, $C_{2,0}$ 비트들을 입력하고 논리곱 연산하여 Carry-in 비트를 출력한다.

<142> h) 상기 CSA2 250의 출력중 MSB로부터의 $(n+1)$ 개의 합 값들과 항 $(n+2)$ 개의 캐리 값들이 CSA1 220의 입력으로 피드백된다. 이때 CSA 1 합 항의 상위 두 비트와 캐리항의 상위 한 비트에는 "0"을 입력으로 하며 2 비트가 오른쪽으로 쉬프트되도록 결선하여 피드백이 이루어진다. 즉, 상기 CSA2 250의 가산기 256으로부터 출력되는 합 값 $S_{2,n+2}$는 CSA1 220의 상위 3번째 가산기 223으로 제공되고, 최상위 및 상위 2번째 가산기들 224,225에는 "0"의 합 값이 제공된다. 상기 CSA2 250의 가산기 256으로부터 출력되는 캐리 값 $C_{2,n+2}$는 CSA1 220의 상위 2번째 가산기 224로 제공되고, 최상위 가산기 225에는 "0"의 캐리 값이 제공된다.

<143> i) $(m+2)$ 클럭 동안에 상기 b)~h) 단계들이 수행된 이후에는 다음과 같은 동작이 수행된다. 즉, CPA(Carry Propagation Adder)(도시하지 않음)는 상기 CSA2 150의 출력인 캐리 값과 합 값에 대해 가산 동작을 수행한다.

<144> 예를 들어, *A*, *B*, *N*이 각각 다음의 <표 11>과 같이 12비트일 때 상기 절차에 의한 Montgomery 모듈러 연산 결과는 <표 12> 및 <표 13>에 나타낸 바와 같다. 이때 최종적인 연산 결과는 다음과 같다. *FinalResult*: 0111.1100.0111(0x7C7)+0010.1000.0000(0x280)+1010.0100.1000(0xA48).

<145>

N=000.1010.0101.1001 (0xA59) B=000.0101.1100.0011 (0x5C3)

2N=001.0100.1011.0010 (0x13B2) 2B=000.1011.1000.0110 (0xB86)

3N=001.1111.0000.1011 (0x1F0B) 3B=001.0001.0100.1001 (0x1149)

【수학식 5】 A=000.1001.0011.1110 (0x93E)

<146> 【표 7】

i	A_i	CSA 1 out S C	Carry_in	S_1S_0
i	0	000.0000.0000.0000 0000.0000.0000.000	0	00
0	2	000.1011.1000.0110 0000.0000.0000.000	0	10
1	3	001.0110.1100.0101 0000.0010.1001.001	0	11
2	3	001.0111.1010.0010 0000.0010.1001.001	1	01
3	0	000.1001.0100.1111 0000.0101.0000.000	1	00
4	1	000.0110.0101.0000 0000.0011.0000.011	1	11
5	2	000.1001.0110.1101 0000.0111.0000.010	1	10
6	0	000.0100.0010.0100 0000.0101.0010.010	1	01
7	0	000.0101.0001.0000 0000.0101.0000.010	1	01

<147>

<147> 1020030016100

출력 일자: 2004/3/24

【표 8】

i	A_i	$S_1 S_0$	$q_1 q_0$	CSA 2 out S C	Carry in
i	0	00	00	000.0000.0000.0000 0000.0000.0000.000	0
0	2	10	10	(0.0).001.1111.0011.0100 (0).0000.0001.0000.010	0
1	3	11	01	(0.0)001.1110.0000.1110 (0).0000.0101.1010.001	1
2	3	01	11	(0.0).000.1010.0011.1010 (0).0010.1111.0000.011	1
3	0	00	00	(0.0)000.1100.0100.1110 (0).0000.0010.0000.001	1
4	1	11	01	(0.0)000.1111.0000.1110 (0).0000.0100.1010.001	1
5	2	10	10	(0.0)001.1010.1101.1010 (0).0000.1010.0100.101	1
6	0	01	11	(0.0)001.1110.0000.1010 (0).0000.1010.0100.101	1
7	0	01	11	(0.0)001.1111.0001.1110 (0).0000.1010.0000.001	1

<148> 다음에, 전술한 바와 같은 본 발명의 장치에 의한 연산 결과 값을 이용하여 모듈라 곱셈 연산 $A \cdot B \bmod N$ 을 계산하는 절차를 설명하기로 한다. 이에 대한 구체적인 하드웨어 구성은 생략되고 있음에 유의하여야 한다.

<149> 1) 미리 $P=2^{2(n+4)} \bmod N$ 을 계산한다.

<150> 2) 본 발명의 장치를 이용하여 $C=A \cdot B \cdot 2^{-(n+4)} \bmod N$ 을 계산한다.

<151> 3) $P \cdot C \cdot 2^{-(n+4)} \bmod N = A \cdot B \bmod N$ 을 계산한다.

<152> 그 다음에, 전술한 바와 같은 본 발명의 장치에 의한 연산 결과 값을 이용하여 RSA 연산 수행에 필요한 모듈라 멱승 $m^e \bmod N$을 계산하는 절차를 설명하기로 한다.

<153> 1) 지수 e를 레지스터(혹은 메모리)에 저장한다.

<154> 2) 레지스터 N에 modulus N을 저장해 둔다.

<155> 3) 임시 레지스터 C와 S를 0으로 초기화한다.

<156> 4) Montgomery 모듈라 곱셈 $m'=fm(m,P,N)=m\cdot P\cdot R^{-1} \bmod N$을 수행한다. 단, 멱승 연산의 밑 P는 위 절차에서 정의한 미리 계산된 값을 나타내며, $R=2^{n+4}$이다.

<157> 5) m'를 레지스터 B에 로딩한다.

<158> 6) 레지스터 B에 로딩한 값을 이용해서 모듈라 제곱 연산을 수행한다. 이때 Montgomery 모듈라 곱셈에 필요한 승수 A는 레지스터 B에서 로딩하며, radix-4 Recoding 회로를 이용해서 값을 얻는다.

<159> 7) 지수 e를 왼쪽으로 쉬프트 한다.

<160> 8) 지수 e의 MSB(Most Significant Bit) 1을 무시하고 다음 비트부터 다음 9) - 10) 단계를 수행한다.

<161> 9) 지수 e의 매 비트가 0 또는 1 에 상관없이 단계 4) -5)를 수행하여 모듈라 제곱 연산을 수행한다. 이때 제곱 연산에 필요한 승수는 레지스터 A, 피승수는 레지스터 B에 저장된다.

<162> 10) 지수 e의 현재 비트가 1인 경우에는 단계 9)을 수행한 후, 단계 4) -5)를 수행하여 모듈라 곱셈 연산을 수행한다. 이때 피승수는 레지스터 B의 내용이고 승수는 멱승 연산의 밑 m'이다.

<163> 11) 지수 e의 모든 비트에 대해서 단계 8) - 10)을 수행한 후 단계 4)를 이용해서 모듈라 곱셈을 한 번 더 수행한다. 이때 피승수는 레지스터 B의 내용이고 승수는 1이다.

<164> 상기 단계 1)- 11)을 수행한 후 레지스터 C와 S에 남아 있는 값에 대해 CPA(Carry Propagation Adder)를 수행한 값이 최종적인 멱승 값 $m^e \bmod N$ 이 된다.

<165> C-4. 발명의 효과

<166> 전술한 바와 같이, 본 발명은 $A \cdot B \cdot 2^{-(n+4)} \bmod N$ 을 계산하는 회로를 기술한 것으로 기술된 회로를 이용하여 상기 일반적인 모듈라 곱셈인 $A \cdot B \bmod N$ 을 계산할 수 있다. 이러한 본 발명에 기초하여 계산된 $A \cdot B \bmod N$ 는 디지털 서명을 생성 및 검증하기 위한 기기에 사용될 수 있는 하드웨어적인 장치에 사용될 수 있다. 또한, 본 발명은 IC 카드를 바탕으로 한 전자 서명, 인증, 암/복호화의 하드웨어 장치에 사용될 수 있다. 또한, 본 발명은 모듈라 곱셈을 수행하는 전자 서명 기기에 의해서 암호 및 복호화 하는 기기를 제공할 수 있다. 또한, 본 발명은 전자 서명 기기를 바탕으로 해서, NIST-DSS, RSA, ElGamal, Schnorr 전자 서명 등 기존의 공개키 암호 시스템을 구현하는데 이용될 수 있다.

<167> D. 발명의 적용 예

<168> 도 13은 본 발명에서 구현한 암호 보조 장치를 이용해서 암호화 및 전자 서명을 할 수 있는 IC 카드의 블록 구성을 나타내는 도면이다.

<169> 상기 도 13에서, 중앙처리장치(CPU: Central Processing Unit) 310은 암호, 인증 및 전자 서명을 가능하게 하는 명령어를 해독하고, 도뮬라 연산 프로세서(modular arithmetic coprocessor) 330에 필요한 제어신호와 데이터를 제공한다. 롬(ROM: Read Only Memory) 350의 내부에는 암호화 및 전자 서명 등에 필요한 키(key)와 같이 보호받아야 하는 데이터를 위한 보안모듈(security module)이 존재한다.

<170> 한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 아니되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

【발명의 효과】

<171> 상술한 바와 같이 본 발명은 본 발명은 $A \cdot B \cdot 2^{-(\omega+4)} \bmod N$ 을 계산하는 회로를 기술한 것으로 기술된 회로를 이용하여 상기 일반적인 모듈라 곱셈인 $A \cdot B \bmod N$을 계산할 수 있다. 이러한 본 발명에 기초하여 계산된 $A \cdot B \bmod N$ 는 디지털 서명을 생성 및 검증하기 위한 기기에 사용될 수 있는 하드웨어적인 장치에 사용될 수 있다. 또한, 본 발명은 IC 카드를 바탕으로 한 전자 서명, 인증, 암/복호화의 하드웨어 장치에 사용될 수 있다. 또한, 본 발명은 모듈라 곱셈을 수행하는 전자 서명 기기에 의해서 암호 및 복호화 하는 기기를 제공할 수 있다. 또한, 본 발명은 전자 서명 기기를 바탕으로 해서, NIST-DSS, RSA, ElGamal, Schnorr 전자 서명 등 기존의 공개키 암호 시스템을 구현하는데 이용될 수 있다.

<169> 상기 도 13에서, 중앙처리장치(CPU: Central Processing Unit) 310은 암호, 인증 및 전자 서명을 가능하게 하는 명령어를 해독하고, 도뮬라 연산 프로세서(modular arithmetic coprocessor) 330에 필요한 제어신호와 데이터를 제공한다. 롬(ROM: Read Only Memory) 350의 내부에는 암호화 및 전자 서명 등에 필요한 키(key)와 같이 보호받아야 하는 데이터를 위한 보안모듈(security module)이 존재한다.

<170> 한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 아니되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

【발명의 효과】

<171> 상술한 바와 같이 본 발명은 본 발명은 $A \cdot B \cdot 2^{-(n+4)} \bmod N$ 을 계산하는 회로를 기술한 것으로 기술된 회로를 이용하여 상기 일반적인 모듈라 곱셈인 $A \cdot B \bmod N$을 계산할 수 있다. 이러한 본 발명에 기초하여 계산된 $A \cdot B \bmod N$ 는 디지털 서명을 생성 및 검증하기 위한 기기에 사용될 수 있는 하드웨어적인 장치에 사용될 수 있다. 또한, 본 발명은 IC 카드를 바탕으로 한 전자 서명, 인증, 암/복호화의 하드웨어 장치에 사용될 수 있다. 또한, 본 발명은 모듈라 곱셈을 수행하는 전자 서명 기기에 의해서 암호 및 복호화 하는 기기를 제공할 수 있다. 또한, 본 발명은 전자 서명 기기를 바탕으로 해서, NIST-DSS, RSA, ElGamal, Schnorr 전자 서명 등 기존의 공개키 암호 시스템을 구현하는데 이용될 수 있다.

【특허청구범위】

【청구항 1】

n비트의 승수(A)와 피승수(B)를 입력하고 $m+2$ 클럭(여기서, $m=n/2$) 내에서 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$) 을 계산하기 위하여 몽고메리 유형의 모듈라 곱셈 연산을 수행하는 장치에 있어서,

상기 승수의 비트들이 순차적으로 쉬프트되어 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 상기 피승수를 다중화하여 $(n+4)$비트의 부호있는 이진 수들을 출력하는 기록 로직과,

$(n+4)$개의 전가산기들을 구비하고, $(n+2)$비트의 제1 신호와, $(n+3)$비트의 제2 신호와, 상기 기록 로직으로부터의 $(n+4)$비트의 상기 이진 수들을 제3 신호로 입력하고, 상기 제1 신호의 상위 번째 비트는 상기 전가산기들중 상위 3개의 전가산기들로 입력되고, 상기 제2 신호의 상위 $(n+3)$번째 비트는 상기 전가산기들중 상위 2개의 전가산기들로 입력되고, 상기 $(n+4)$개의 전가산기들에 의해 $(n+4)$비트의 캐리 값들과 합 값들을 출력하는 제1 캐리저장형 가산기(CSA)와,

상기 제1 CSA로부터의 상기 $(n+4)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들과 하위 1개의 전가산기로부터 출력되는 캐리 값을 입력하고, 모듈라 감소의 배수를 결정하기 위한 3비트의 결정 값을 출력하는 몫 로직과,

상기 결정 값에 따라 미리 정해진 모듈로 수의 집합중 하나의 모듈로 수를 선택하여 출력하는 선택기와,

(n+4)개의 전가산기들을 구비하고, 상기 선택기로부터의 선택된 (n+4)비트의 모듈로 수를 제1 신호로서 입력하고, 상기 제1 CSA로부터의 상기 (n+4)비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 (n+3)비트의 캐리 값들을 제2 신호로서 입력하고, 상기 (n+4)비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 (n+3)비트의 합 값들을 제3 신호로서 입력하고, 상기 제1 신호의 (n+4)비트들은 상기 전가산기들의 최하위 전가산기들로부터 순차 입력되고, 상기 제2 신호의 (n+3)비트들은 상기 전가산기들중 하위 2번째 전가산기들로부터 순차 입력되고, 상기 제3 신호의 (n+3)비트들은 상기 전가산기들의 하위 2번째 전가산기들로부터 순차 입력되고, 상기 (n+4)개의 전가산기들에 의해 (n+4)비트의 캐리 값들과 합 값들을 출력하는 제2 캐리저장형 가산기(CSA)와,

상기 제2 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 하위 2번째 전가산기로부터 출력되는 합 값을 전가산하여 캐리 입력 값으로 상기 몫 로직으로 제공하는 전가산기와,

상기 제2 CSA로부터의 캐리 값들과 합 값들을 가산하여 출력하는 캐리전파형 가산기(CPA)를 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 2】

제1항에 있어서,

상기 승수의 비트들을 순차적으로 쉬프트시켜 상기 쉬프트된 비트열을 생성하는 쉬프트 레지스터와,

상기 피승수를 저장하기 위한 레지스터를 더 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 3】

제1항에 있어서, 상기 기록 로직은,

상기 생성된 비트열의 하위 2비트들을 부스 기록하는 기록회로와,

상기 부스 기록된 결과와 상기 피승수를 다중화하는 다중화기와,

상기 생성된 비트열의 하위 2비트들에 따라 상기 다중화기의 출력을 선택적으로 1의 보수화하고 상기 $(n+4)$비트의 부호있는 이진 수들을 출력하는 보수화기를 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 4】

제1항에 있어서, 상기 몫 로직은,

상기 캐리 입력값과 상기 제1 CSA의 최하위 전가산기로부터 출력되는 합 값을 전가산하는 전가산기와,

상기 제1 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 2번째 전가산기로부터 출력되는 합 값을 배타적 논리합하는 배타적 논리합기와,

상기 전가산기와 상기 배타적 논리합기의 출력들과 미리 설정된 입력비트를 조합하여 상기 3비트의 결정 값으로 출력하는 조합회로를 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 5】

제4항에 있어서, 상기 캐리 입력값을 저장하기 위한 디플립플롭을 더 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 6】

제4항에 있어서, 상기 몫 로직의 상기 전가산기 및 상기 배타적 논리합기로는 미리 설정된 보정용 캐리 값이 제공됨을 특징으로 하는 상기 곱셈 장치.

【청구항 7】

제6항에 있어서, 상기 몫 로직의 상기 전가산기로는 상기 피승수의 부호 비트가 제공됨을 특징으로 하는 상기 곱셈 장치.

【청구항 8】

제4항에 있어서, 상기 몫 로직의 상기 전가산기의 출력은 상기 제2 CSA의 최하위 전가산기로 제공됨을 특징으로 하는 상기 곱셈 장치.

【청구항 9】

n비트의 승수(A)와 피승수(B)를 입력하고 $m+2$ 클럭(여기서, $m=n/2$) 내에서 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m-2}$)을 계산하기 위하여 몽고메리 유형의 모듈라 곱셈 연산을 수행하는 장치에 있어서,

상기 승수의 비트들이 순차적으로 쉬프트되어 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 상기 피승수를 다중화하여 $(n+3)$비트의 이진 수들을 출력하는 기록 로직과,

$(n+3)$개의 전가산기들을 구비하고, $(n+1)$비트의 제1 신호와, $(n+2)$비트의 제2 신호와, 상기 기록 로직으로부터의 $(n+4)$비트의 상기 이진 수들을 제3 신호로 입력하고, 상기 제1 신호의 상위 $(n+1)$번째 비트는 상기 전가산기들중 상위 3번째의 전가산기로 입력되고, 상기 제2 신호의 상위 $(n+2)$번째 비트는 상기 전가산기들중 상위 2번째의 전가산기로 입력되고, 상기 전가산기들중 최상위 전가산기로 "0"레벨의 상기 제1 신호 및 상기 제2 신호가 입력되고, 상기 전가산기들중 상위 2번째의 전가산기로 "0"레벨의 상기 제1 신호가 입력되고, 상기 $(n+3)$개의 전가산기들에 의해 $(n+3)$비트의 캐리 값들과 합 값들을 출력하는 제1 캐리저장형 가산기(CSA)와,

상기 제1 CSA로부터의 상기 $(n+3)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들과 하위 1개의 전가산기로부터 출력되는 캐리 값을 입력하고, 모듈라 감소의 배수를 결정하기 위한 2비트의 결정 값을 출력하는 룩 로직과,

상기 결정 값에 따라 미리 정해진 모듈로 수의 집합중 하나의 모듈로 수를 선택하여 출력하는 선택기와,

$(n+3)$개의 전가산기들을 구비하고, 상기 선택기로부터의 선택된 $(n+3)$비트의 모듈로 수를 제1 신호로서 입력하고, 상기 제1 CSA로부터의 상기 $(n+3)$비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 $(n+2)$비트의 캐리 값들을 제2 신호로서 입력하고, 상기 $(n+3)$비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 $(n+2)$비트의 합 값들을 제3 신호로서 입력하고,

상기 제1 신호의 (n+3)비트들은 상기 전가산기들의 최하위 전가산기들로부터 순차 입력되고, 상기 제2 신호의 (n+2)비트들은 상기 전가산기들중 하위 2번째 전가산기들로부터 순차 입력되고, 상기 제3 신호의 (n+2)비트들은 상기 전가산기들의 하위 2번째 전가산기들로부터 순차 입력되고,. 상기 (n+3)개의 전가산기들에 의해 (n+4)비트의 캐리 값들과 합 값들을 출력하는 제2 캐리저장형 가산기(CSA)와,

상기 제2 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 하위 2번째 전가산기로부터 출력되는 합 값을 논리곱 연산하여 캐리 입력 값으로 상기 몫 로직으로 제공하는 논리곱 연산기와,

상기 제2 CSA로부터의 캐리 값들과 합 값들을 가산하여 출력하는 캐리전파형 가산기(CPA)를 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 10】

제9항에 있어서,

상기 승수의 비트들을 순차적으로 쉬프트시켜 상기 쉬프트된 비트열을 생성하는 쉬프트 레지스터와,

상기 피승수를 저장하기 위한 레지스터를 더 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 11】

제9항에 있어서, 상기 기록 로직은, 상기 생성된 비트열의 하위 2비트들에 따라 상기 피승수를 다중화하여 출력하는 다중화기임을 특징으로 하는 상기 곱셈 장치.

【청구항 12】

제9항에 있어서, 상기 몫 로직은,

상기 캐리 입력값과 상기 제1 CSA의 최하위 전가산기로부터 출력되는 합 값을 반가산하는 반가산기와,

상기 제1 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 2번째 전가산기로부터 출력되는 합 값을 배타적 논리합하는 배타적 논리합기와,

상기 반가산기와 상기 배타적 논리합기의 출력들과 미리 설정된 입력비트를 조합하여 상기 2비트의 결정 값으로 출력하는 조합회로를 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 13】

제12항에 있어서, 상기 캐리 입력값을 저장하기 위한 디플립플롭을 더 포함함을 특징으로 하는 상기 곱셈 장치.

【청구항 14】

제12항에 있어서, 상기 몫 로직의 상기 반가산기의 출력은 상기 제2 CSA의 최하위 전가산기로 제공됨을 특징으로 하는 상기 곱셈 장치.

【청구항 15】

n비트의 승수(A)와 피승수(B)를 입력하고 $m+2$ 클럭(여기서, $m=n/2$) 내에서 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하기 위하여 몽고메리 유형의 모듈라 곱셈 연산을 수행하는 방법에 있어서,

상기 승수의 비트들을 순차적으로 쉬프트하여 쉬프트된 비트열을 생성하는 과정과,

상기 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 상기 피승수를 다중화하여 $(n+4)$비트의 부호있는 이진 수들을 출력하는 과정과,

$(n+4)$개의 전가산기들을 구비하는 제2 캐리저장형 가산기(CSA)에 $(n+2)$비트의 제1 신호와, $(n+3)$비트의 제2 신호와, 상기 기록 로직으로부터의 $(n+4)$비트의 상기 이진 수들을 제3 신호로 입력하고, 상기 제1 신호의 상위 $(n+2)$번째 비트를 상기 전가산기들중 상위 3개의 전가산기들에 입력하고, 상기 제2 신호의 상위 $(n+3)$번째 비트를 상기 전가산기들중 상위 2개의 전가산기들에 입력하고, 상기 제1 캐리저장형 가산기(CSA)의 상기 $(n+4)$개의 전가산기들에 의해 $(n+4)$비트의 캐리 값들과 합 값들을 출력하는 과정과,

상기 제1 CSA로부터의 상기 $(n+4)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들과 하위 1개의 전가산기로부터 출력되는 캐리 값을 입력하고, 모듈라 감소의 배수를 결정하기 위한 3비트의 결정 값을 출력하는 과정과,

상기 결정 값에 따라 미리 정해진 모듈로 수의 집합중 하나의 모듈로 수를 선택하여 출력하는 과정과,

$(n+4)$개의 전가산기들을 구비하는 제2 캐리저장형 가산기(CSA)에 상기 선택기로부터의 선택된 $(n+4)$비트의 모듈로 수를 제1 신호로서 입력하고, 상기 제1 CSA로부터의 상기 $(n+4)$비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 $(n+3)$비트의 캐리 값들을 제2 신호로서 입력하고, 상기 $(n+4)$비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 $(n+3)$비트의 합 값들을 제3 신호로서 입력하고, 상기 제1 신호의 $(n+4)$비트들을 상기 전가산기들의 최하위 전가산기들로부터 순차 입력하고, 상기 제2 신호의 $(n+3)$비트들을 상기 전가산기들중 하위 2번째 전가산기들로부터 순차 입력하고, 상기 제3 신호의 $(n+3)$비트들을 상기 전가산기들의 하위 2번째 전가산기들로부터 순차 입력하고, 상기 $(n+4)$개의 전가산기들에 의해 $(n+4)$비트의 캐리 값들과 합 값들을 출력하는 과정과,

상기 제2 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 하위 2번째 전가산기로부터 출력되는 합 값을 전가산하여 캐리 입력 값으로 상기 몫 로직으로 제공하는 과정과,

상기 제2 CSA로부터의 캐리 값들과 합 값들을 가산하여 출력하는 과정을 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 16】

제15항에 있어서, 상기 부호있는 이진 수들을 출력하는 과정은,

상기 생성된 비트열의 하위 2비트들을 부스 기록하는 단계와,

상기 부스 기록된 결과와 상기 피승수를 다중화하는 단계와,

상기 생성된 비트열의 하위 2비트들에 따라 상기 다중화된 출력을 선택적으로 1의 보수화하고 상기 (n+4)비트의 부호있는 이진 수들을 출력하는 단계를 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 17】

제15항에 있어서, 상기 3비트의 결정 값을 출력하는 과정은,

상기 캐리 입력값과 상기 제1 CSA의 최하위 전가산기로부터 출력되는 합 값을 전가산하는 단계와,

상기 제1 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 2번째 전가산기로부터 출력되는 합 값을 배타적 논리합하는 단계와,

상기 전가산 단계에서의 결과와 상기 배타적 논리합 단계에서의 결과와 미리 설정된 입력비트를 조합하여 상기 3비트의 결정 값으로 출력하는 단계를 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 18】

제17항에 있어서, 상기 캐리 입력값을 일시적으로 저장하는 단계를 더 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 19】

제17항에 있어서, 상기 배타적 논리합 단계에서의 연산 동작을 위해 미리 설정된 보정용 캐리 값이 제공되는 단계를 더 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 20】

제17항에 있어서, 상기 전가산 단계에서의 연산 동작을 위해 미리 설정된 보정용 캐리 값과 상기 피승수의 부호 비트가 제공되는 단계를 더 포함함을 징으로 하는 상기 곱셈 방법.

【청구항 21】

n비트의 승수(A)와 피승수(B)를 입력하고 $m+2$클럭(여기서, $m=n/2$) 내에서 $A \cdot B \cdot R^{-1} \bmod N$ (여기서, $R=4^{m+2}$)을 계산하기 위하여 몽고메리 유형의 모듈라 곱셈 연산을 수행하는 방법에 있어서,

상기 승수의 비트들을 순차적으로 쉬프트하여 쉬프트된 비트열을 생성하는 과정과,

상기 생성된 비트열의 하위 2비트들을 부스 기록(Booth recording)하고 상기 부스 기록된 결과와 상기 피승수를 다중화하여 $(n+3)$비트의 이진 수들을 출력하는 과정과,

$(n+3)$개의 전가산기들을 구비하는 제1 캐리저장형 가산기(CSA)로 $(n+1)$비트의 제1 신호와, $(n+2)$비트의 제2 신호와, 상기 기록 로직으로부터의 $(n+4)$비트의 상기 이진 수들을 제3 신호로 입력하고, 상기 제1 신호의 상위 $(n+1)$번째 비트를 상기 전가산기들중 상위 3번째의 전가산기로 입력하고, 상기 제2 신호의 상위 $(n+2)$번째 비트를 상기 전가산기들중 상위 2번째의 전가산기로 입력하고, 상기 전가산기들중 최상위 전가산기로 "0"레벨의 상기 제1 신호 및

상기 제2 신호를 입력하고, 상기 전가산기들중 상위 2번째의 전가산기로 "0"레벨의 상기 제1 신호를 입력하고, 상기 $(n+3)$개의 전가산기들에 의해 $(n+3)$비트의 캐리 값들과 합 값들을 출력하는 과정과,

상기 제1 CSA로부터의 상기 $(n+3)$비트의 캐리 값들과 합 값들중 선택된 하위 2개의 전가산기들로부터 출력되는 합 값들과 하위 1개의 전가산기로부터 출력되는 캐리 값을 입력하고, 모듈라 감소의 배수를 결정하기 위한 2비트의 결정 값을 출력하는 과정과,

상기 결정 값에 따라 미리 정해진 모듈로 수의 집합중 하나의 모듈로 수를 선택하여 출력하는 과정과,

$(n+3)$개의 전가산기들을 구비하는 제2 캐리저장형 가산기(CSA)로 상기 선택기로부터의 선택된 $(n+3)$비트의 모듈로 수를 제1 신호로서 입력하고, 상기 제1 CSA로부터의 상기 $(n+3)$비트의 캐리 값들중 최상위 캐리 값을 제외한 나머지의 $(n+2)$비트의 캐리 값들을 제2 신호로서 입력하고, 상기 $(n+3)$비트의 합 값들중 최하위 캐리 값을 제외한 나머지의 $(n+2)$비트의 합 값들을 제3 신호로서 입력하고, 상기 제1 신호의 $(n+3)$비트들을 상기 전가산기들의 최하위 전가산기들로부터 순차 입력하고, 상기 제2 신호의 $(n+2)$비트들을 상기 전가산기들중 하위 2번째 전가산기들로부터 순차 입력하고, 상기 제3 신호의 $(n+2)$비트들을 상기 전가산기들의 하위 2번째 전가산기들로부터 순차 입력하고, 상기 $(n+3)$개의 전가산기들에 의해 $(n+4)$비트의 캐리 값들과 합 값들을 출력하는 과정과,

상기 제2 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 하위 2번째 전가산기로부터 출력되는 합 값을 논리곱 연산하여 캐리 입력 값으로 상기 몫 로직으로 제공하는 과정과,

상기 제2 CSA로부터의 캐리 값들과 합 값들을 가산하여 출력하는 과정을 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 22】

제21항에 있어서, 상기 2비트의 결정 값을 출력하는 과정은,

상기 캐리 입력값과 상기 제1 CSA의 최하위 전가산기로부터 출력되는 합 값을 반가산하는 단계와,

상기 제1 CSA의 최하위 전가산기로부터 출력되는 캐리 값과 2번째 전가산기로부터 출력되는 합 값을 배타적 논리합하는 단계와,

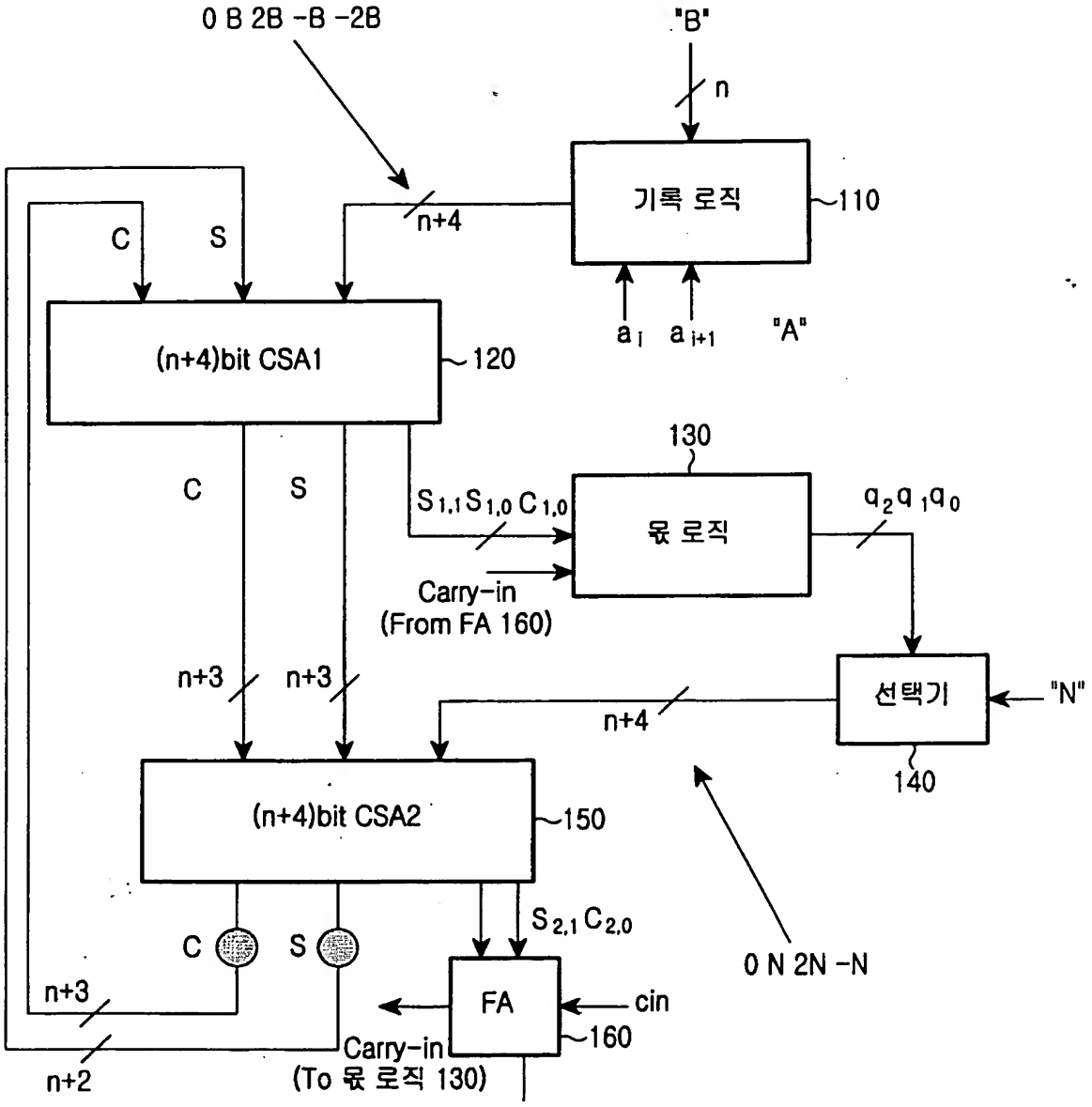
상기 반가산 단계에서의 결과와 상기 배타적 논리합 단계에서의 결과와 미리 설정된 입력비트를 조합하여 상기 2비트의 결정 값으로 출력하는 단계를 포함함을 특징으로 하는 상기 곱셈 방법.

【청구항 23】

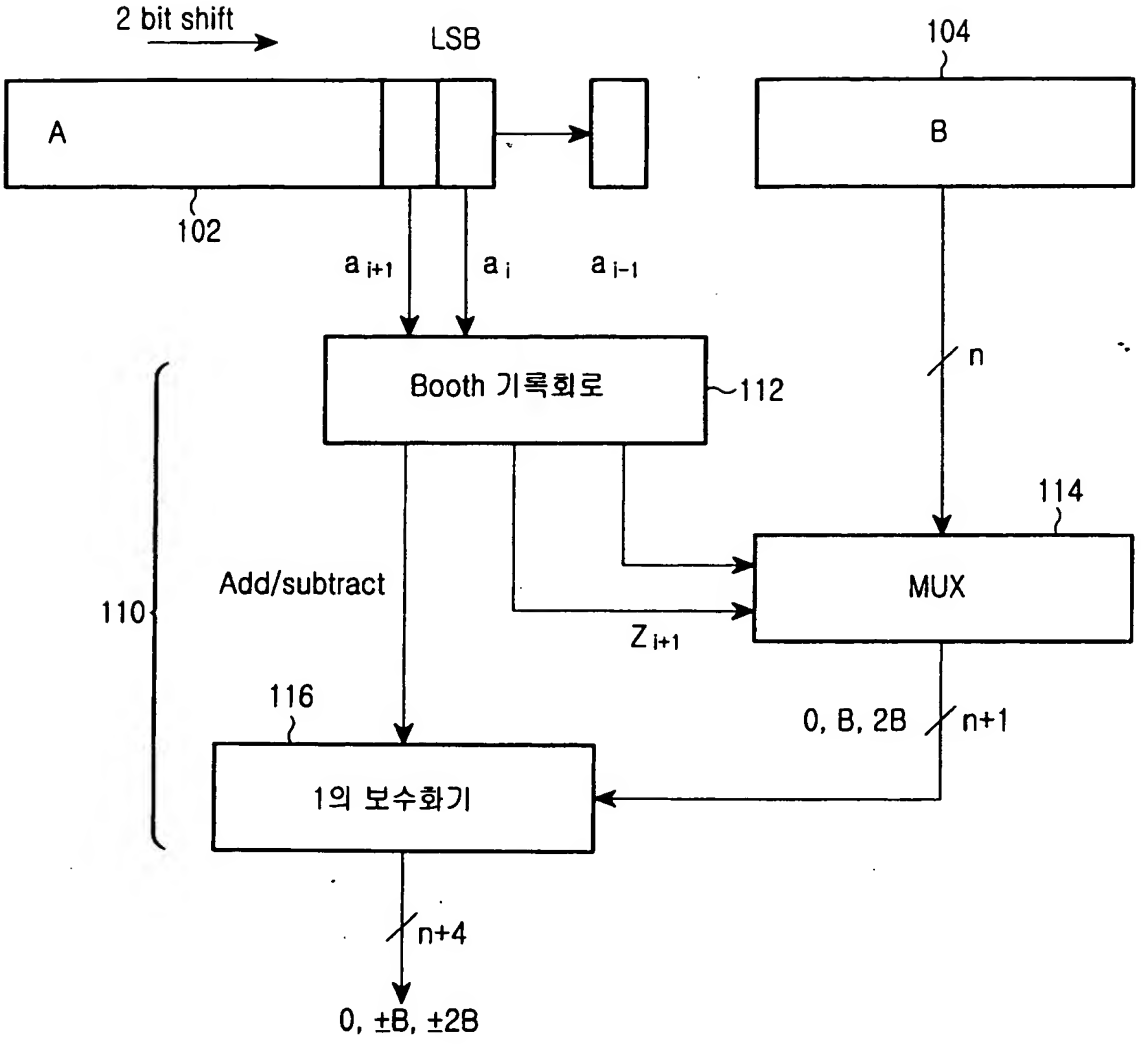
제22항에 있어서, 상기 캐리 입력값을 일시적으로 저장하는 단계를 더 포함함을 특징으로 하는 상기 곱셈 방법.

【도면】

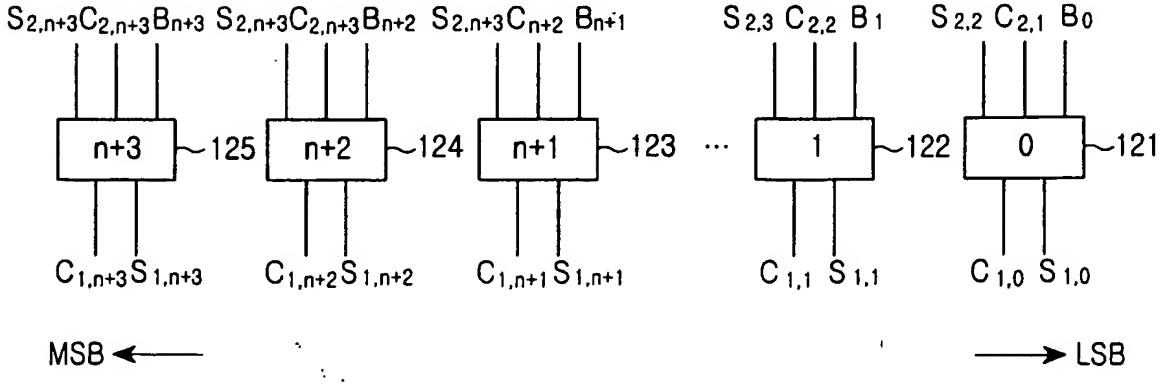
【도 1】


0 B 2B -B -2B
"B"
n
기록 로직
110
n+4
C
S
(n+4)bit CSA1
120
a_i a_{i+1}
"A"
130
C
S
$S_{1,1}S_{1,0}C_{1,0}$
룹 로직
$q_2q_1q_0$
Carry-in
(From FA 160)
n+3
n+3
선택기
"N"
n+4
140
(n+4)bit CSA2
150
$S_{2,1}C_{2,0}$
C
S
0 N 2N -N
n+3
FA
cin
160
Carry-in
(To 룹 로직 130)
n+2

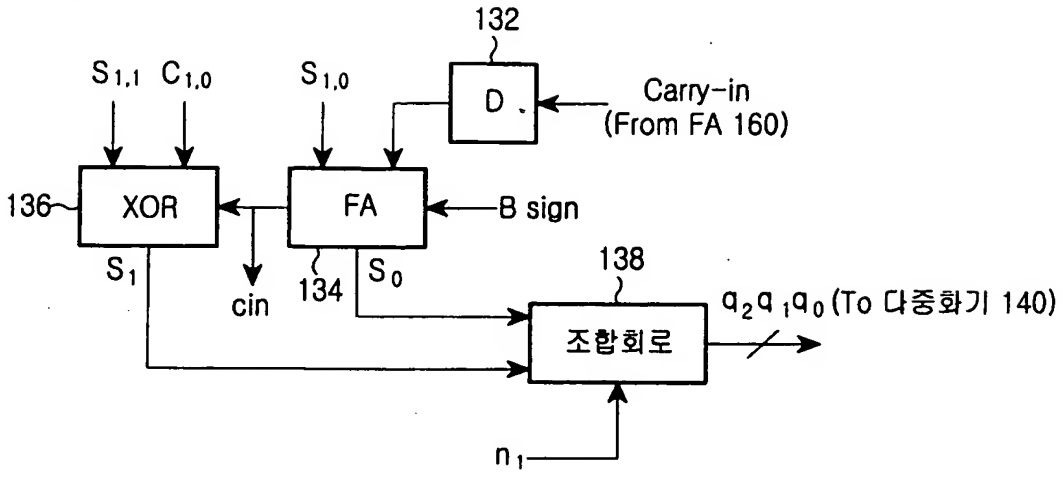
【도 2】


2 bit shift
LSB
104
A
B
102
a_{i+1}
a_i
a_{i-1}
Booth 기록회로
112
n
114
Add/subtract
MUX
110
Z_{i+1}
116
0, B, 2B
n+1
1의 보수화기
n+4
0, ±B, ±2B

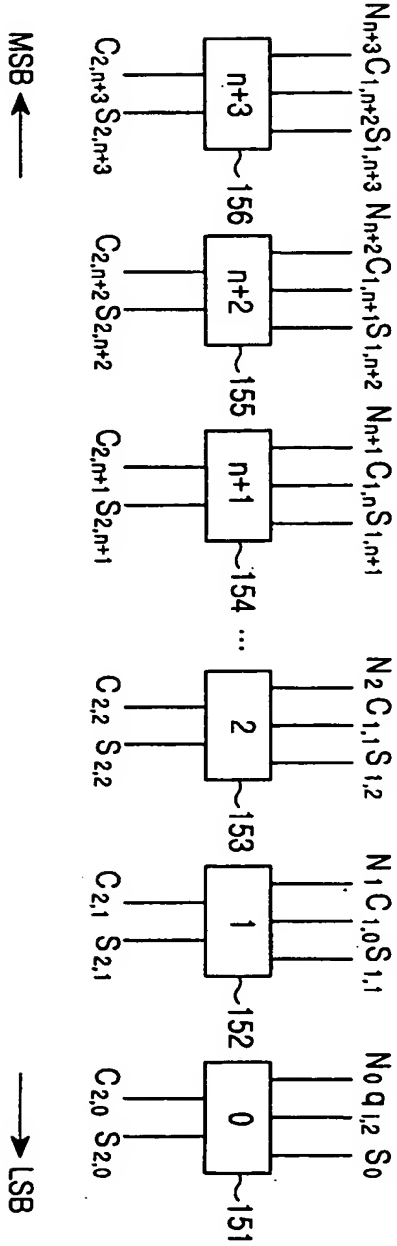
【도 3】


$S_{2,n+3}C_{2,n+3}B_{n+3}$
$S_{2,n+3}C_{2,n+3}B_{n+2}$
$S_{2,n+3}C_{n+2}$ B_{n+1}
$S_{2,3}$ $C_{2,2}$ B_1
$S_{2,2}$ $C_{2,1}$ B_0
n+3
125
n+2
124
n+1
123
1
122
0
121
$C_{1,n+3}S_{1,n+3}$
$C_{1,n+2}S_{1,n+2}$
$C_{1,n+1}S_{1,n+1}$
$C_{1,1}$ $S_{1,1}$
$C_{1,0}$ $S_{1,0}$
MSB
LSB

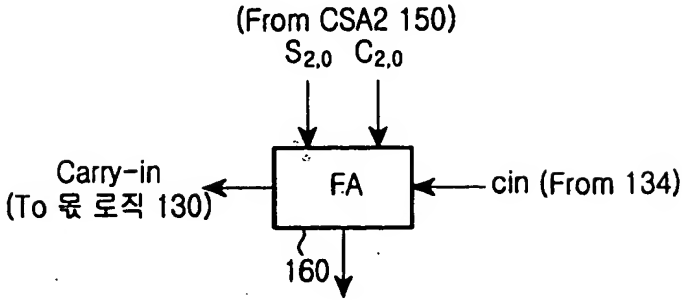
【도 4】


132
$S_{1,1}$ $C_{1,0}$
$S_{1,0}$
D
Carry-in
(From FA 160)
136
XOR
FA
B sign
S_1
cin
134
S_0
138
$q_2q_1q_0$ (To 다중화기 140)
조합회로
n_1

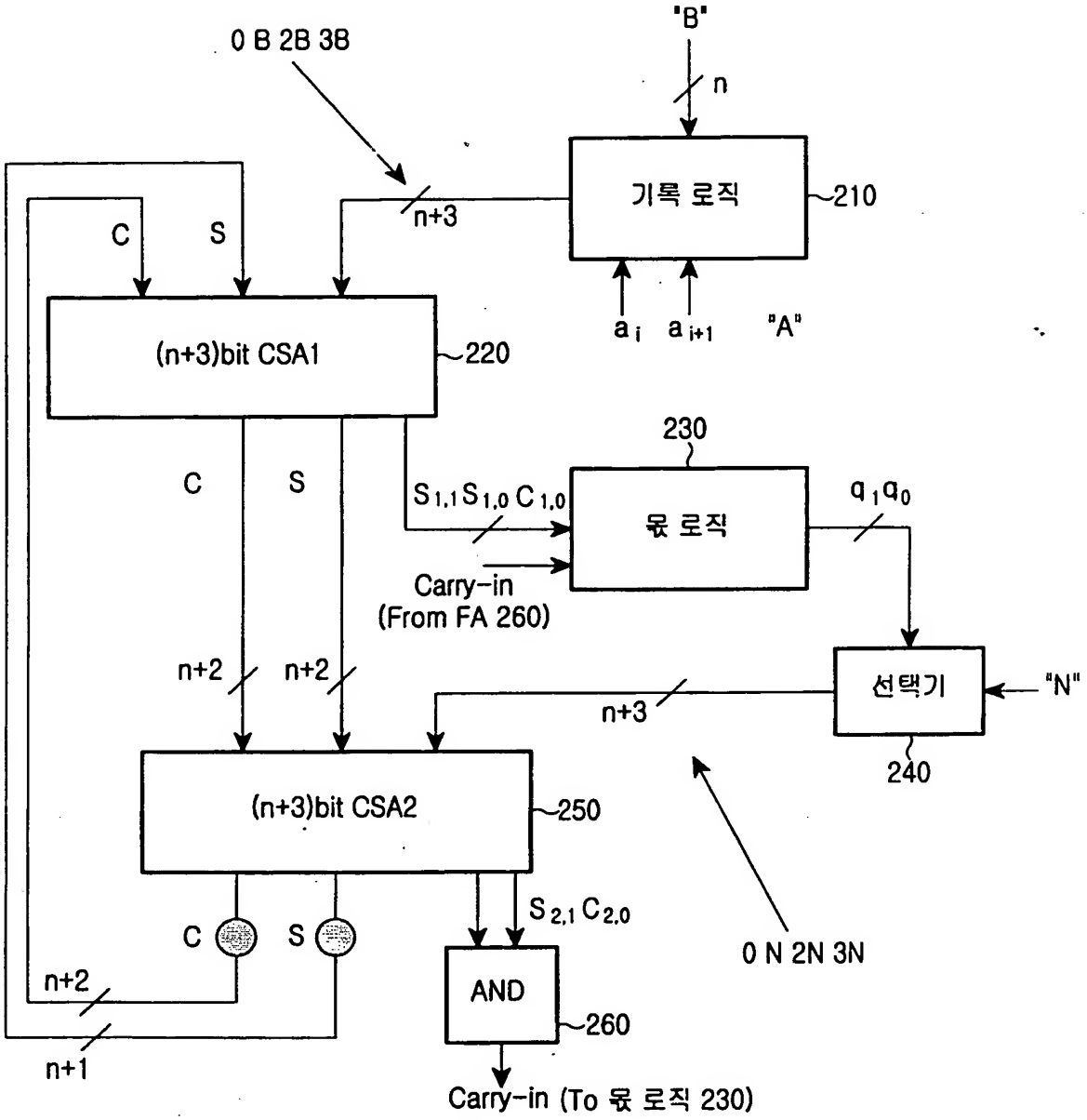
【도 5】


MSB
LSB
n+3
n+2
n+1
2
1
0
156
155
154
153
152
151
$C_{2,n+3}$ $S_{2,n+3}$
N_{n+3} $C_{1,n+2}$ $S_{1,n+3}$
$C_{2,n+2}$ $S_{2,n+2}$
N_{n+2} $C_{1,n+1}$ $S_{1,n+2}$
$C_{2,n+1}$ $S_{2,n+1}$
N_{n+1} $C_{1,n}$ $S_{1,n+1}$
$C_{2,2}$ $S_{2,2}$
N_2 $C_{1,1}$ $S_{1,2}$
$C_{2,1}$ $S_{2,1}$
N_1 $C_{1,0}$ $S_{1,1}$
$C_{2,0}$ $S_{2,0}$
N_0 $C_{1,2}$ S_0

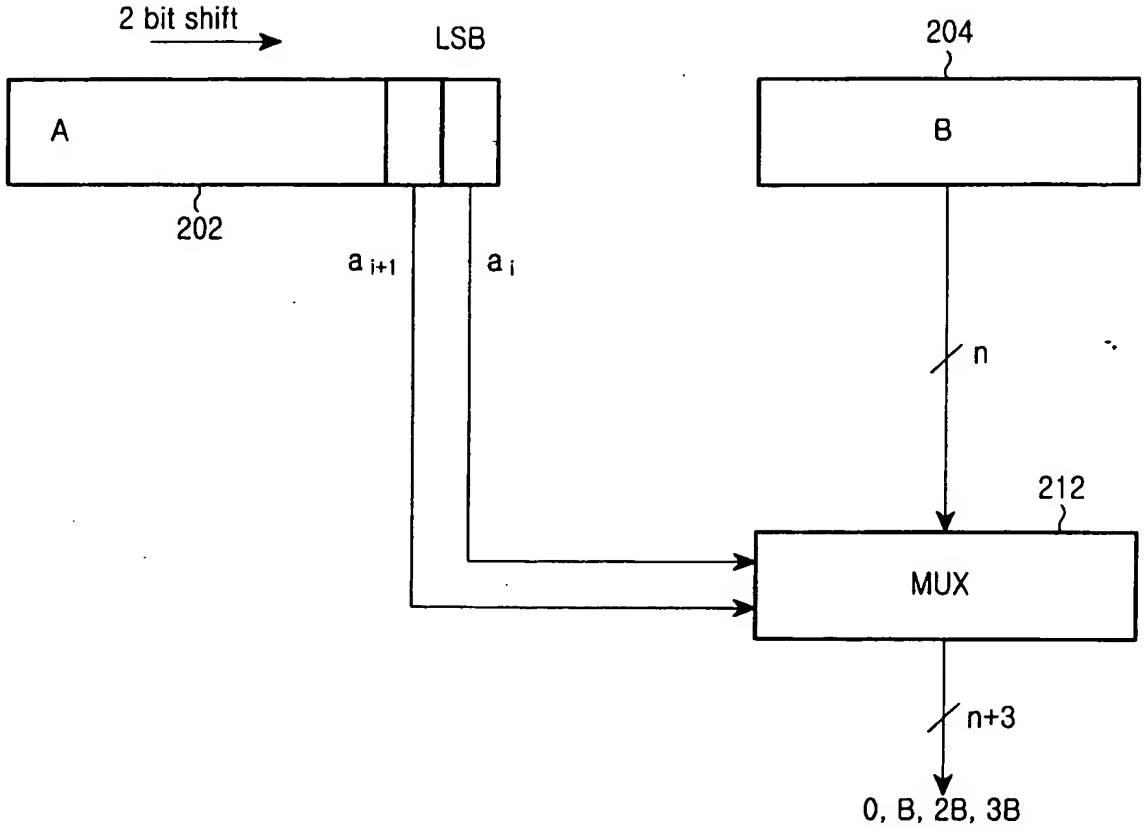
【도 6】


(From CSA2 150)
$S_{2,0}$ $C_{2,0}$
Carry-in
(To 맵 모듈 130)
FA
cin (From 134)
160

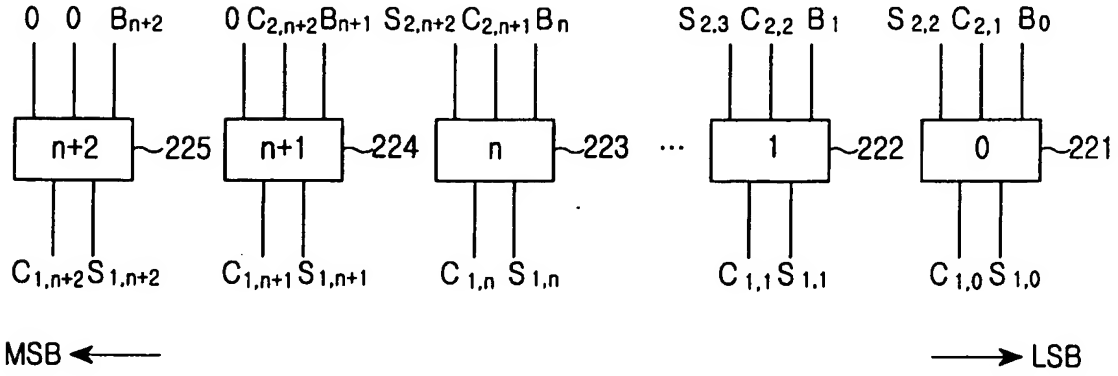
【도 7】


0 B 2B 3B
"B"
n
기록 로직
210
n+3
C
S
(n+3)bit CSA1
220
a_i a_{i+1} "A"
230
C
S
$S_{1,1}S_{1,0}C_{1,0}$
룩 로직
q_1q_0
Carry-in
(From FA 260)
n+2
n+2
선택기
"N"
n+3
240
(n+3)bit CSA2
250
$S_{2,1}C_{2,0}$
C
S
0 N 2N 3N
n+2
AND
260
n+1
Carry-in (To 룩 로직 230)

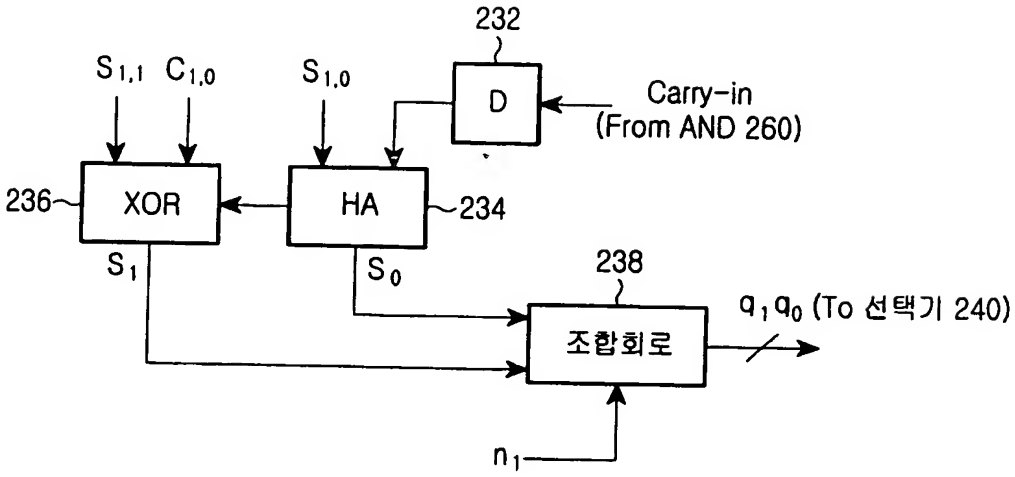
【도 8】


2 bit shift
LSB
204
A
B
202
a_{i+1}
a_i
n
212
MUX
n+3
0, B, 2B, 3B

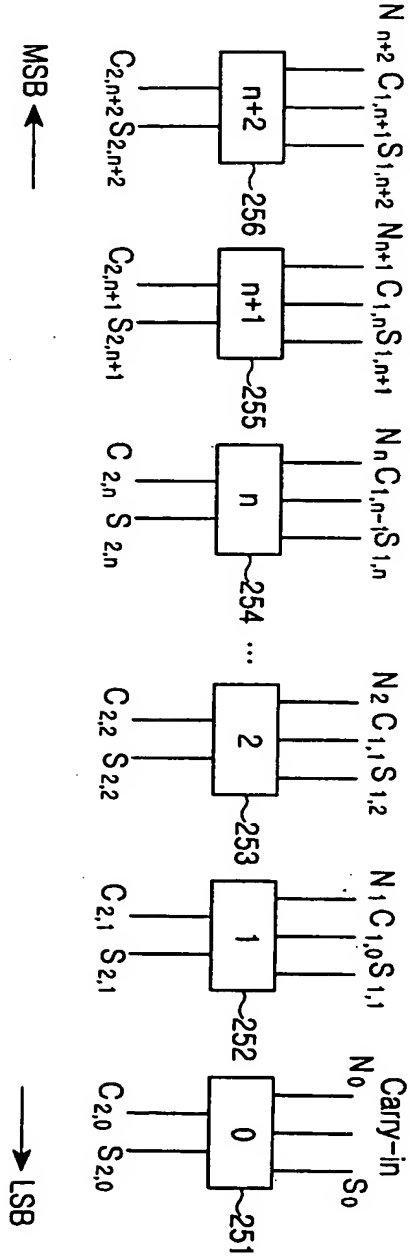
【도 9】


0 0 B_{n+2}
0 $C_{2,n+2}$ B_{n+1}
$S_{2,n+2}$ $C_{2,n+1}$ B_n
$S_{2,3}$ $C_{2,2}$ B_1
$S_{2,2}$ $C_{2,1}$ B_0
n+2
225
n+1
224
n
223
1
222
0
221
$C_{1,n+2}$ $S_{1,n+2}$
$C_{1,n+1}$ $S_{1,n+1}$
$C_{1,n}$ $S_{1,n}$
$C_{1,1}$ $S_{1,1}$
$C_{1,0}$ $S_{1,0}$
MSB
LSB

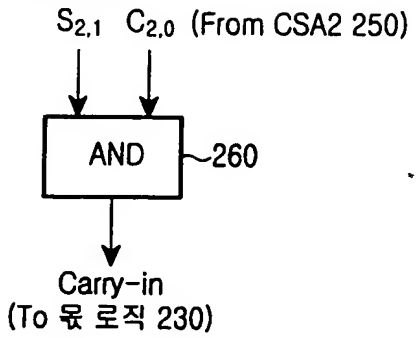
【도 10】


232
$S_{1,1}$ $C_{1,0}$
$S_{1,0}$
D
Carry-in
(From AND 260)
236
XOR
HA
234
S_1
S_0
238
q_1q_0 (To 선택기 240)
조합회로
n_1

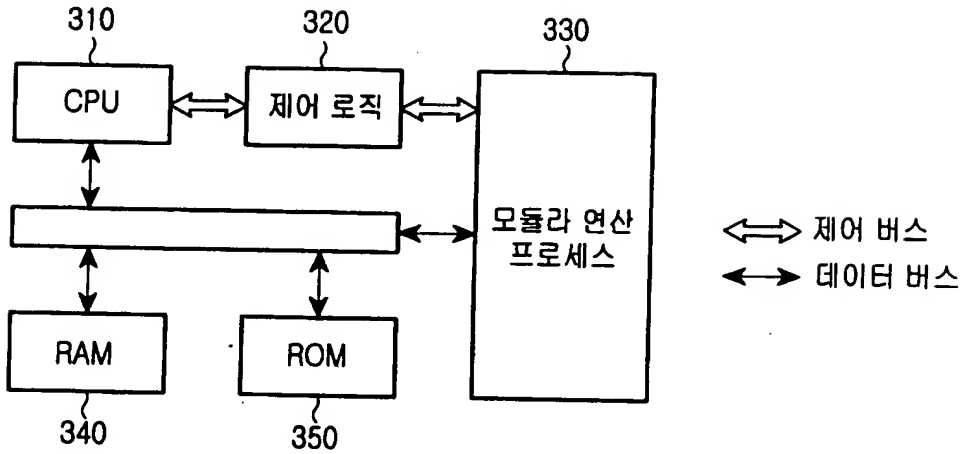
【도 11】


MSB
LSB
n+2
n+1
n
2
1
0
256
255
254
253
252
251
Carry-in
N_{n+2} $C_{1,n+1}$ $S_{1,n+2}$
$C_{2,n+2}$ $S_{2,n+2}$
N_{n+1} $C_{1,n}$ $S_{1,n+1}$
$C_{2,n+1}$ $S_{2,n+1}$
N_n $C_{1,n-1}$ $S_{1,n}$
$C_{2,n}$ $S_{2,n}$
N_2 $C_{1,1}$ $S_{1,2}$
$C_{2,2}$ $S_{2,2}$
N_1 $C_{1,0}$ $S_{1,1}$
$C_{2,1}$ $S_{2,1}$
N_0 S_0
$C_{2,0}$ $S_{2,0}$

【도 12】


$S_{2,1}$ $C_{2,0}$ (From CSA2 250)
AND
260
Carry-in
(To 몫 로직 230)

【도 13】


310
320
330
CPU
제어 로직
모듈라 연산 프로세스
RAM
ROM
340
350
제어 버스
데이터 버스